Actividad 2 - Aplicación de análisis de sentimientos

October 24, 2023

1 Actividad 2: Aplicación de análisis de sentimientos

Alfonso Pineda Cedillo | A01660394

Fecha de entrega: 27 de Octubre de 2023

1.1 Instrucciones

- Seleccione un corpus a trabajar.
  - Puede ser el corpus utilizado en los ejercicios anteriores.
  - Un dataset de tweets podría ser un buen ejemplo.
  - Dataset minado de alguna red social como todos los comentarios de un post de una figura pública.
- Aplicar un modelo pre entrenado del "sentiment analisis" y analizar reacciones de un post.

1.2 Solución

Como primer paso, se importan las librerías necesarias para el desarrollo de la actividad.

[60]:
```
import pandas as pd
from transformers import TFAutoModelForSequenceClassification, AutoTokenizer,␣
 ↪AutoConfig
import numpy as np
from scipy.special import softmax
import re
import urllib.request
import csv
```

Posteriormente, realizamos la carga de nuestro dataset, el cual contiene 1,600,000 tweets extraídos usando el API de Twitter. El dataset contiene 6 columnas, de las cuales solo nos interesa la columna de `text` y la columna `target`, la cual indica si el tweet es positivo, negativo o neutral. Para este proceso, ocupamos la librería `pandas` y la función `read_csv` para leer el archivo.

[6]:
```
# Cargar el conjunto de datos desde el archivo CSV
DATASET_COLUMNS=['target','ids','date','flag','user','text']
dataset = pd.read_csv('datasets/tweets.csv', encoding='ISO-8859-1',␣
 ↪names=DATASET_COLUMNS)
```

Asimismo, realizamos un pequeño análisis exploratorio de los datos, para ver el formato general de la base de datos y ver si hay datos faltantes. Para esto, ocupamos la función `head()` para ver los primeros 5 registros de la base de datos, y la función `info()` para ver el tipo de dato de cada columna y si hay datos faltantes.

[8]:
```
dataset.head()
```

[8]:
```
   target         ids                          date      flag  \
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                                                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…
```

[65]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1600000 entries, 0 to 1599999
Data columns (total 6 columns):
 #   Column  Non-Null Count    Dtype
---  ------  --------------    -----
 0   target  1600000 non-null  int64
 1   ids     1600000 non-null  int64
 2   date    1600000 non-null  object
 3   flag    1600000 non-null  object
 4   user    1600000 non-null  object
 5   text    1600000 non-null  object
dtypes: int64(2), object(4)
memory usage: 73.2+ MB
```

Para probar el modelo pre-entrenado, ocupamos únicamente la columna `text`, la cual contiene el contenido del tweet. Para esto, almacenamos los datos de la columna en una variable llamada `data` y posteriormente, observamos su contenido.

[40]:
```
data = dataset[['text']]
data
```

[40]:
```
                                                   text
0       @switchfoot http://twitpic.com/2y1zl - Awww, t…
1       is upset that he can't update his Facebook by …
2       @Kenichan I dived many times for the ball. Man…
```

```
3          my whole body feels itchy and like its on fire
4        @nationwideclass no, it's not behaving at all…
…                                                      …
1599995  Just woke up. Having no school is the best fee…
1599996  TheWDB.com - Very cool to hear old Walt interv…
1599997  Are you ready for your MoJo Makeover? Ask me f…
1599998  Happy 38th Birthday to my boo of alll time!!! …
1599999  happy #charitytuesday @theNSPCC @SparksCharity…

[1600000 rows x 1 columns]
```

A continuación, realizamos la carga del modelo pre-entrenado y del tokenizador. Para esto, se define la ruta al modelo pre-entrenado que se desea utilizar. El modelo en cuestión se llama "cardiffnlp/twitter-roberta-base-sentiment". Este modelo ha sido pre-entrenado en datos relacionados con Twitter y está diseñado para realizar tareas de análisis de sentimiento en texto de Twitter.

Asimismo, ocupamos la función `AutoTokenizer.from_pretrained` para cargar el tokenizador. El tokenizador es una parte esencial del procesamiento de texto, ya que se encarga de dividir el texto en unidades más pequeñas, como palabras o subpalabras, que son comprensibles para el modelo. En este caso, el tokenizador se adapta al modelo pre-entrenado y está diseñado específicamente para procesar texto de Twitter.

Luego, se carga la configuración del modelo utilizando la función AutoConfig.from_pretrained(MODEL). La configuración del modelo incluye detalles sobre la arquitectura y las configuraciones específicas del modelo, como el número de capas, el tamaño del embedding, entre otros. Esta información es esencial para asegurarse de que el modelo se utilice de manera adecuada.

Finalmente, se carga el modelo en sí utilizando la función TFAutoModelForSequenceClassification.from_pretrained(MODEL). El modelo pre-entrenado es una red neuronal profunda que ha sido entrenada en una gran cantidad de datos para realizar tareas de clasificación de secuencias, en este caso, análisis de sentimiento. El modelo ya ha aprendido patrones y características relevantes en el lenguaje natural a partir de los datos de Twitter.

[54]:
```
# Cargar el modelo preentrenado y el tokenizador
MODEL = f"cardiffnlp/twitter-roberta-base-sentiment"
tokenizer = AutoTokenizer.from_pretrained(MODEL)
config = AutoConfig.from_pretrained(MODEL)
model = TFAutoModelForSequenceClassification.from_pretrained(MODEL)
```

```
Downloading (…)lve/main/config.json:   0%|          | 0.00/747 [00:00<?, ?B/s]

Downloading (…)olve/main/vocab.json:   0%|          | 0.00/899k [00:00<?, ?B/s]

Downloading (…)olve/main/merges.txt:   0%|          | 0.00/456k [00:00<?, ?B/s]

Downloading (…)cial_tokens_map.json:   0%|          | 0.00/150 [00:00<?, ?B/s]

Downloading tf_model.h5:   0%|          | 0.00/501M [00:00<?, ?B/s]

All model checkpoint layers were used when initializing
TFRobertaForSequenceClassification.
```

```
All the layers of TFRobertaForSequenceClassification were initialized from the
model checkpoint at cardiffnlp/twitter-roberta-base-sentiment.
If your task is similar to the task the model of the checkpoint was trained on,
you can already use TFRobertaForSequenceClassification for predictions without
further training.
```

Posteriormente, se define una función denominada `preprocess` con el propósito de realizar el preprocesamiento de texto. Este procedimiento es fundamental en el procesamiento de lenguaje natural y se utiliza para limpiar y estandarizar el texto de entrada antes de realizar análisis más avanzados. La función se encarga de aplicar dos principales transformaciones al texto ingresado como argumento.

En primer lugar, la función efectúa la eliminación de menciones de usuario en el texto. Esto se logra mediante el uso de expresiones regulares (regex). Las menciones de usuario en plataformas de redes sociales como Twitter suelen comenzar con el símbolo '@', seguido de un nombre de usuario que puede consistir en letras, números y guiones bajos. La función `re.sub` se utiliza para reemplazar todas las menciones de usuario en el texto original por la etiqueta '@user'. Esta acción es relevante ya que permite eliminar información específica de usuarios en el texto, lo que puede ser útil en tareas de análisis de sentimiento y procesamiento de texto en general.

En segundo lugar, la función lleva a cabo la eliminación de enlaces (URLs) del texto de entrada. Las URLs suelen comenzar con "http" o "https" y pueden ser seguidas por caracteres diversos que representan una dirección web. Al igual que en el caso de las menciones de usuario, la función `re.sub` se utiliza para reemplazar todas las URLs en el texto original por la etiqueta 'http'. Esta acción es significativa porque permite suprimir enlaces web que no suelen contener información relevante para el análisis de sentimiento o procesamiento de texto en sí.

[55]:
```python
# Función para preprocesar el texto
def preprocess(text):
    # Eliminar menciones de usuario
    text = re.sub(r'@[A-Za-z0-9_]+', '@user', text)
    # Eliminar enlaces
    text = re.sub(r'http\S+', 'http', text)
    return text
```

A continuación, definimos una función llamada `analyze_sentiment`, que se utiliza para realizar el análisis de sentimiento en un tweet dado. Esta función realiza una serie de pasos para evaluar la polaridad del sentimiento (positivo, negativo o neutral) en el texto del tweet.

En primer lugar, el código invoca la función antes mencionada, llamada `preprocess(text)`. Después del preprocesamiento, se codifica el texto limpio utilizando un tokenizador previamente cargado. El texto procesado se pasa al modelo pre-entrenado para realizar la predicción de sentimiento. El modelo genera una puntuación (score) para cada clase de sentimiento (positivo, negativo y neutral).

Para determinar la etiqueta de sentimiento correspondiente, el código realiza una serie de pasos adicionales. Primero, se normaliza las puntuaciones mediante una función softmax, lo que produce una distribución de probabilidad sobre las clases de sentimiento. Luego, las etiquetas se recuperan a partir de un archivo de mapeo (mapping) ubicado en una URL específica. Las etiquetas representan

las clases de sentimiento a las que se asocian las puntuaciones.

Finalmente, se ordenan las puntuaciones en orden descendente, lo que permite identificar la etiqueta de sentimiento más probable. Las etiquetas y las puntuaciones se imprimen en la consola, lo que brinda información sobre el sentimiento estimado del tweet, junto con la confianza asociada a esa estimación.

[61]:
```
# Función para analizar el sentimiento de un tweet
def analyze_sentiment(text):
    text = preprocess(text)
    encoded_input = tokenizer(text, return_tensors='tf')
    output = model(encoded_input)
    scores = output[0][0].numpy()
    scores = softmax(scores)
    ranking = np.argsort(scores)
    ranking = ranking[::-1]

    labels=[]
    mapping_link = f"https://raw.githubusercontent.com/cardiffnlp/tweeteval/
 ↪main/datasets/sentiment/mapping.txt"
    with urllib.request.urlopen(mapping_link) as f:
        html = f.read().decode('utf-8').split("\n")
        csvreader = csv.reader(html, delimiter='\t')
    labels = [row[1] for row in csvreader if len(row) > 1]

    for i in range(scores.shape[0]):
        label = labels[ranking[i]]
        score = scores[ranking[i]]
        print(f"{i+1}) {label} {np.round(float(score), 4)}")
```

Por último, obtenemos 10 tweets aleatorios del dataset previamente importado y aplicamos la función `analyze_sentiment` a cada uno de ellos. Esto nos permite observar el funcionamiento del modelo pre-entrenado en la práctica y ver cómo se comporta en diferentes casos de uso.

El dataset escogido para probar el modelo es diferente que el utilizado para su entrenamiento. Esto due hecho a propósito para ver cómo se comporta el modelo en un dataset que no ha visto antes. Los resultados obtenidos se imprimen primero mostrando el tweet analizado y posteriormente, la etiqueta de sentimiento y la confianza asociada a esa estimación.

[64]:
```
# Elegir 10 tweets aleatorios
random_tweets = data.sample(10)

# Aplicar el análisis de sentimiento a los 10 tweets aleatorios
for index, row in random_tweets.iterrows():
    text = row['text']
    print(f"Tweet: {text}")
    analyze_sentiment(text)
    print("----------------------------\n")
```

```
Tweet: did some morning edits, should read a bit better now  http://tr.im/n1i5
1) neutral 0.5139
2) positive 0.3661
3) negative 0.12
------------------------------

Tweet: @stellargirl my laptop does both of those things - and crashes FireFox on
a regular basis too
1) negative 0.7873
2) neutral 0.1913
3) positive 0.0215
------------------------------

Tweet: @mollysusie cool. I'll check it out soon. Although I'll be going on
vacation soon so I'll be doing my own hiking around beantown.
1) positive 0.937
2) neutral 0.0615
3) negative 0.0015
------------------------------

Tweet: McAfee is the biggest hog of memory I have ever seen.  Every morning it
slows the work computers.  The loss of productivity must be huge.
1) negative 0.9119
2) neutral 0.0785
3) positive 0.0096
------------------------------

Tweet: @theendishere Thanks 4the follow
1) positive 0.9373
2) neutral 0.0606
3) negative 0.0021
------------------------------

Tweet: missin` already my highschool days…
1) neutral 0.4687
2) negative 0.4222
3) positive 0.1091
------------------------------

Tweet: This rain is killing me…that's y my wedding have to be inside so much
for a outside wedding. .
1) negative 0.926
2) neutral 0.0659
3) positive 0.0081
------------------------------

Tweet: @modulation   Now I always told u u could flip pancakes! And make the
batter for that matter!   Chris X
```

```
1) neutral 0.5525
2) positive 0.3987
3) negative 0.0488
------------------------------

Tweet: @ZacharyQuinto happy birthday, zachary quinto.
1) positive 0.9738
2) neutral 0.025
3) negative 0.0012
------------------------------

Tweet: Offing the phone to reduce risks at SATs. I'm not nervous though  wish me
luck
1) positive 0.7117
2) neutral 0.2665
3) negative 0.0218
------------------------------
```