Modelos de lenguaje de gran tamaño (LLM)

Clasificador de sentimientos

Informe de proyecto

Alumno: Christian Canaan Castro Botek *30/05/2024*

Índice general

1. Introducción

En este trabajo se propone utilizar un modelo del lenguaje grande para resolver un problema de clasificación de sentimientos en reseñas de productos. Este proceso implica categorizar el texto presente en las reseñas como positivo, negativo o neutro. Reduciremos el problema a uno de clasificación binaria, positivo o negativo. Para ello utilizaremos el modelo de BERT (Bidirectional Encoder Representation from Transformers) obteniendo sus pesos preentrenados disponibles en HugginFace, facilitandonos la implementación y la etapa de fine-tuning.

2. Desarrollo

Describiremos los pasos de desarrollo para el proyecto haciendo uso de Colab, un servicio alojado de Jupyter Notebook que ofrece acceso gratuito a recursos de computación

2.1. Configuración de entorno

Realizamos la instalación de las bibliotecas necesarias para el proyecto

```
!pip install -U transformers datasets evaluate accelerate
!pip install tensorboard tensorboardX
!pip install transformers
!pip install transformers[torch]
!pip install accelerate -U
!pip install datasets
!pip install torch
!pip install scikit-learn
```

2.2. Carga del dataset y exploración

Cargamos el dataset [1] directamente desde HuggingFace. El dataset es extenso, por lo que se decidió tomar un número bajo de muestras para empezar a hacer las pruebas de entrenamiento. En una primera instancia se consideró arbitrariamente pero teniendo en cuenta el costo computacional del entrenamiento 10000 muestras aleatorias para construir los dataset de train y test.

Un registro del dataset tiene la siguiente forma:

```
{'label': 1, 'title': 'Works as expected', 'content': "I've had no problems running this with my Dell 8200 PC running Windows XP Home Edition which is running an addon USB 2 card, to which this hub (I have the 8 port version) is connected. When I bought it a year ago, I thought that 8 ports would be overkill and that 4 would be adequate, but I bought this unit anyway. Boy was I glad, right now 6 of the 8 are in use."}
```

2.3. Definición del modelo, tokenizado y tensores

Como fue mencionado, para este proyecto utilizamos el modelo pre-entrenado de BERT [2], el cual creamos y lo definimos con **dos etiquetas** ya que vamos a reducir el problema de clasificación a uno de tipo binario.

El modelo de BERT requiere que las entradas sean convertidas en tokens, lo que implica una representación numérica para el texto crudo. En este caso, realizamos

un tokenizado con tamaño **max_length** para que todas las secuencias sean de igual longitud. Este tokenizado debemos hacerlo para cada entrada del dataset por lo que hacemos un map por lotes para implementarlo. Un ejemplo de registro:


```
{'label': 0, 'title': 'Author seems mentally unstable', 'content':
'I know that Tom Robbins has a loyal following and I started the
book with high expectations. However, I did not enjoy this book as
it was too much work to follow his confused logic. I think that he
was under the influence during most of time that he wrote.',
'input_ids': [101, 1045, 2113, 2008, 3419, 18091, 2038, 1037, 8884,
2206, 1998, 1045, 2318, 1996, 2338, 2007, 2152, 10908, 1012, 2174,
1010, 1045, 2106, 2025, 5959, 2023, 2338, 2004, 2009, 2001, 2205,
2172, 2147, 2000, 3582, 2010, 5457, 7961, 1012, 1045, 2228, 2008,
2002, 2001, 2104, 1996, 3747, 2076, 2087, 1997, 2051, 2008, 2002,
2626, 1012, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0], 'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], 'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]}
```


Una vez realizado el tokenizado y dado que el modelo BERT opera con tensores, debemos preparar los datos de entrada al modelo [5] para que sean tensores. Realizado este paso finalmente dividimos la entrada en un 80% dataset de entrenamiento y un 20% para la evaluación.

2.4. Trainer y entrenamiento

Para el trainer [3] definimos una serie de parámetros iniciales definidas en training_args:

- Establecemos el número de épocas en 3 para realizar las primeras pruebas iniciales del modelo con pocos registros del dataset, esto dado que existieron algunos inconvenientes con los tiempos de ejecución con GPU disponible en el entorno de COLAB.
- Los tamaños del batch para train y eval fueron seteados en 8, también para ser en una primera instancia cuidadosos con el costo computacional.
- Se definió un warmup_steps de 500 para ayudar a estabilizar el entrenamiento en los inicios y evitar pivoteos grandes en los pesos del modelo.
- Se estableció weight_decay en 0.01 como valor típico inicial.

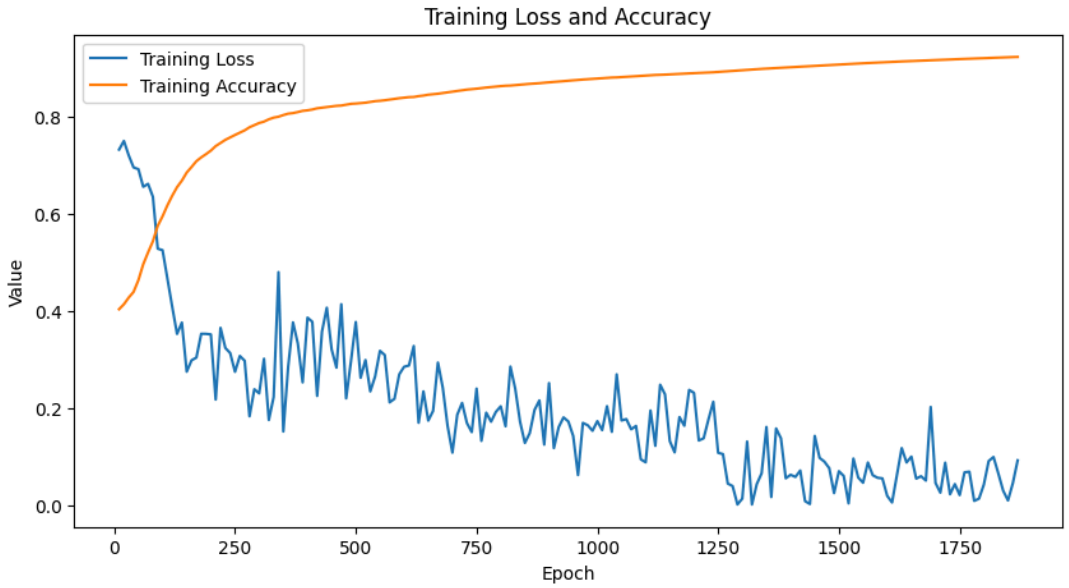
El trainer también cuenta con una función de cálculo de métricas para posterior análisis sobre el comportamiento del modelo.

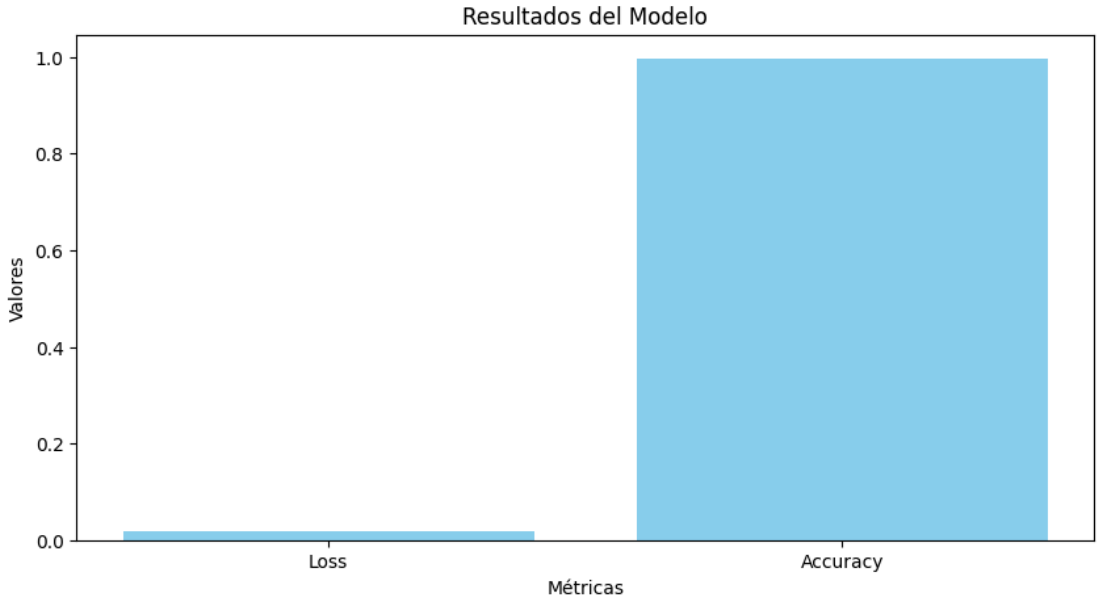
2.5. Evaluación

Finalmente se realiza una evaluación del modelo entrenado [4], se grafican y visualizan algunos resultados de las métricas calculadas haciendo uso de la librería matplotlib.

3. Resultados

Luego del entrenamiento y la evaluación las métricas reflejan un resultado aceptable





Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.327800	0.138676	0.957500	0.957648	0.966801	0.948667
2	0.108000	0.049754	0.985500	0.985636	0.989066	0.982231
3	0.092500	0.018071	0.996500	0.996547	0.996055	0.997038

Resultados para las 3 epochs realizadas

Estos resultados [8] son puestos a prueba poniendo al modelo en modo evaluación y catalogando los resultados de alimentarlo con nuevas reviews generadas manualmente. Finalmente se observa que es capaz de clasificar el sentimiento predominante en cada review.

Review: This product is amazing! I love it and will definitely buy again.
Predicted Sentiment: Positive

Review: Terrible experience. The product broke after one use.
Predicted Sentiment: Negative

Review: Not bad, but could be better. The quality is okay.
Predicted Sentiment: Positive

Review: Excellent quality and fast shipping. Highly recommend this!
Predicted Sentiment: Positive

Review: Waste of money. I'm very disappointed.
Predicted Sentiment: Negative

Review: Absolutely fantastic! Exceeded my expectations in every way.
Predicted Sentiment: Positive

Review: Horrible service, and the product arrived damaged.
Predicted Sentiment: Negative

Review: Pretty decent for the price. Satisfied with the purchase.
Predicted Sentiment: Positive

Review: I am thrilled with this product! It's exactly what I needed.
Predicted Sentiment: Positive

Review: The worst purchase I've ever made. Completely useless.
Predicted Sentiment: Negative

Review: Superb quality and very durable. Will buy more in the future.
Predicted Sentiment: Positive

4. Conclusiones

En este trabajo, hemos implementado y entrenado un modelo BERT para la clasificación de sentimientos en reseñas de productos de Amazon. Los resultados muestran que el modelo alcanza un buen rendimiento en términos del accuracy, demostrando la eficacia de los modelos de lenguaje grande para tareas de procesamiento del lenguaje natural y logrando predicciones exitosas cuando el modelo fue expuesto a nuevas reviews.

Resulta evidente que se pueden lograr mejoras haciendo mayores ajustes sobre los hiperparámetros, o añadiendo capas adicionales. Esto siempre y cuando se disponga de la capacidad de cómputo.

5. Referencias

[1] Amazon_polarity dataset, Hugging Face, 2024, Dataset.
[2] BERT, Hugging Face, 2024, BERT Documentation.
[3] Trainer, 2024, Trainer Documentation
[4] How Transformers Work, Josep Ferrer, 2024, A Detailed Exploration of Transformer Architecture.
[5] What is BERT?, Javier Canales Luna, 2023, An Intro to BERT Models.
[6] Introducción a LLM, Josep Ferrer, 2024, Guía introductoria para perfeccionar los LLM.
[7] Fine-Tuning, Sovit Rath, 2023, Fine-Tuning BERT using Hugging Face Transformers.
[8] Métricas del conjunto de entrenamiento, mbforbes, 2021, Metrics for Training Set in Trainer