En la práctica anterior de clasificación de tweets con descripciones de imágenes, se consiguieron buenos resultados utilizando XML-Roberta, aunque había mucho margen de mejora. Para este proyecto, se ha optado por partir del trabajo realizado previamente, mejorando y adaptando el modelo a esta nueva tarea.

En primer lugar, fue necesario arreglar el algoritmo de búsqueda de hiperparámetros utilizado anteriormente, ya que siempre devolvía la primera configuración probada como la mejor. La causa de dicho error radicaba en que el conteo del modelo no se reiniciaba correctamente. En la práctica anterior, que contaba con un dataset más amplio, el bug no tuvo un impacto significativo, dado que el número de pruebas posibles por hora era de apenas cuatro. Por otro lado, en este caso, con un dataset mucho más pequeño, el número de combinaciones que se pueden probar es considerablemente mayor, y el error tuvo más relevancia.

Una vez solucionado el problema del script de entrenamiento, se debía decidir cuál iba a ser el modelo y el tokenizador a usar para esta tarea. Observé que, en la tarea anterior, los resultados más destacados se obtuvieron utilizando un modelo GPT no especificado. Asimismo, otro grupo consiguió muy buenos resultados empleando un tokenizador especializado en Twitter. A partir de estas observaciones, parecía razonable utilizar un modelo GPT junto con tiktoken (un tokenizador especializado en datos de TikTok desarrollado por OpenAI) para esta tarea de clasificación. Sin embargo, este enfoque presentaba varios inconvenientes. El primero de ellos es que tiktoken está diseñado para trabajar únicamente con textos en inglés, aunque podría ser útil para etiquetar estos textos. El segundo problema es que los modelos GPT, como GPT-4 o GPT-3.5, no pueden ajustarse localmente, ya que son privativos. Aunque existen otros modelos como GPT-4All, estos no son compatibles con dicho tokenizador, por lo que se descartó esta opción.

Después de realizar un análisis más profundo de los datos del problema, me di cuenta de que no era necesario contar con un tokenizador especializado en TikTok, ya que los textos en el dataset están en inglés y en español estándar. Para mejorar los resultados obtenidos en la práctica anterior, se decidió evaluar si había alguna mejora al separar los datos.

En este laboratorio, se optó por probar SpanBerta, un modelo ajustado de Hugging Face basado en Roberta, pero especializado en español. Lamentablemente, no fue posible utilizar el modelo en su versión "large", ya que, según las especificaciones, no cabía completamente en memoria. Para los datos en inglés, se utilizó Roberta-large, que tiene la ventaja de ocupar casi lo mismo que XML-Roberta base, pero con entrenamiento exclusivamente en inglés.

Los resultados de los mejores experimentos tras realizar una hora de búsqueda, utilizando la misma configuración que en el laboratorio anterior, fueron los siguientes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Dataset | Batch Size | Epoch | Learning rate | Loss |
| XML-Roberta-base | ALL | 4 | 3 | 1.01e-05 | 0.59 |
| XML-Roberta-base | ES | 8 | 5 | 4.19e-05 | 0.56 |
| Span-Berta-base | ES | 32 | 3 | 1.24e-05 | 0.70 |
| XML-Roberta-base | EN | 8 | 5 | 3.47e-05 | 0.58 |
| Roberta-large | EN | 4 | 4 | 1.36e-05 | 0.52 |

Los experimentos indican que el mejor modelo para clasificar en español es XML-Roberta utilizando únicamente los datos en español, mientras que para los datos en inglés, el mejor modelo es Roberta-large. Por tanto, para obtener los mejores resultados, se optó por combinar las predicciones de ambos modelos especializados. No obstante, también se enviarán al concurso las predicciones de XML-Roberta-base-ALL y las de XML-Roberta-base-ALL+Roberta-EN, ya que en el concurso se pueden enviar tres ficheros de resultados.

En conclusión, utilizar datasets separados para cada idioma otorga ligeras mejoras, incluso si se utiliza el mismo modelo base. Sin embargo, la diferencia radica más en la elección de un checkpoint adecuado para cada idioma.