Parte 3: Fine-tuning

Primeros pasos...

1. Asegúrate de haber iniciado el ambiente mt_finetuning: `conda activate mt_finetuning`
2. Entra a la carpeta FineTuning_Lab_Activity: `cd FineTuning_Lab_Activity`
3. Verás dos archivos: test_set.txt y train_and_test_model.py

test_set.txt es una lista de oraciones que abarcan varias temas. Las mismas servirán como índice de la calidad de traducción. A lo largo de esta actividad, verás cómo las traducciones de las oraciones cambian según las características lingüísticas de los datos que se usan para refinar (fine-tune) un modelo pre-entrenado (pre-trained model).

train_and_test_model.py es el programa que descarga un modelo pre-entrenado y lo refina según un cuerpo de datos especificado. El programa requiere 4 argumentos: tu nombre de usuario, el cuerpo de datos para refinar el modelo, las oraciones de prueba, y la cantidad de ciclos (epochs) que se usan para entrenar al modelo. Un ciclo corresponde a una evaluación de todo el modelo. Típicamente, cuanto más ciclos hay más representativo es el modelo de los datos de entrenamiento. El cuerpo de datos consta de la información lingüística con sus características estilísticas, las cuales influirán en las traducciones que produce el modelo refinado. He seleccionado tres cuerpos que representan tres perspectivas distintas:

- books: Un cuerpo de novelas y otros tipos de literatura del proyecto OPUS. Son libros que han entrado en el dominio común. Pueden ver la lista completa de los libros en español también.
- bible: El texto completo de la biblia en varios idiomas, también del proyecto OPUS.
- science: Un cuerpo de traducciones paralelas de varios artículos científicos del proyecto scielo en América Latina. Puedes aprender más del cuerpo de datos en su paper.

También tienen la opción de indicar 'base' como cuerpo. En este caso, el programa sólo descargará el modelo original y adivinará las traducciones de las oraciones de prueba. En este caso no será necesario indicar el número de ciclos (porque no hay).

Por ejemplo, si quisiera refinar el modelo con datos científicos y un de ciclo entrenamiento, ejecutaría lo siguiente: `python train_and_test_model.py -u gberry03 -d science -t test_set.txt -e 1` Si necesitas más ayuda, puedes ejecutar lo siguiente: `python train_and_test_model.py --help`

A probar (100 puntos; 20 puntos cada uno)

1. Crea una lista de traducciones según el modelo original: `python train_and_test_model.py -u MYUSERNAME -d base -t test_set.txt`. Los resultados se guardarán en una carpeta según tu nombre de usuario.
2. Mira los resultados con vim, emacs, nano, u otra herramienta, por ejemplo: `vim gberry03/base_results.txt`. Puedes copiar y pegar los resultados a un documento local si quieres.
3. Empezando con un ciclo de evaluación, refina el model con los tres cuerpos de datos y evalúa el model refinado con las oraciones de prueba.
4. Repite el proceso de refinamiento utilizando 3 ciclos de evaluación
5. Elige uno de los cuerpos de datos y varía la cantidad de épocas metódicamente: 1, 2, 5, 10.
6. Asegúrate de que los archivos se guardaron en la carpeta correspondiente. Copia los resultados a un documento en tu máquina local.

Reflexión (100 puntos)

1. Comenta sobre la calidad de las traducciones del modelo pre-entrenado.
2. ¿Cómo afecta el refinamiento a la calidad de traducción? Ofrece ejemplos concretos.
3. ¿Qué efecto tiene el número de ciclos de evaluación en la calidad de la traducción? Ofrece ejemplos concretos.
4. ¿Qué modelo te parece ser el mejor? ¿Cuáles son sus especificaciones? ¿Por qué es mejor que los otros?