Diego Alejandro González Vargas – 202110240
Nicolas Bedoya Figueroa – 202212100
Santiago Martínez Novoa – 202112020
Nicolas Rozo Fajardo – 202112920

INFORME DE IMPLEMENTACIÓN – TAREA 5

1. INTRODUCCIÓN

En este informe se presenta un resumen de las implementaciones desarrolladas en los notebooks, cuyo objetivo fue construir y evaluar clasificadores para dos conjuntos de datos utilizando modelos con arquitecturas *encoder* y *decoder*, con el fin de comparar su rendimiento en la tarea específica. En particular, se entrenaron clasificadores para los conjuntos de datos *20 Newsgroup* (20 clases) y *Multi-Domain Sentiment Analysis* (2 clases), empleando los modelos *encoder bert-base-uncased* y *distilbert-base-uncased*, así como los modelos *decoder* de código abierto *GPT-2, GPT-2-Xl* y *Gemma2*. El siguiente análisis aborda la comparación de desempeño, robustez y características prácticas de la implementación de ambas arquitecturas, junto con una discusión sobre aspectos específicos de su implementación.

2. DETALLES DE LA IMPLEMENTACIÓN

La carga y preparación de los conjuntos de datos fue común para ambas arquitecturas. Los conjuntos de datos se obtuvieron a partir de sus archivos fuente y se almacenaron en *dataframes*, los cuales se dividieron en tres subconjuntos: entrenamiento, validación y prueba. Los subconjuntos de entrenamiento y validación se utilizaron exclusivamente en los modelos *encoder* durante la etapa de *fine-tunning*, mientras que el conjunto de prueba se empleó para la evaluación final de todos los modelos, tanto *encoder* como *decoder*, calculando sobre él las métricas de desempeño. Cabe resaltar que esta decisión se tomo ya que lo modelos *decoder* no se re-entrenaron y había que garantizar que ambas arquitecturas fueran evaluadas sobre el mismo conjunto de datos para que los resultados fueran comparables.

A todos los textos se les aplicó la misma función de limpieza, encargada de eliminar direcciones de correo electrónico, enlaces y otros elementos textuales que no son relevantes para la tarea de clasificación y que, por el contrario, aumenta el tamaño de la entrada y la demanda de recursos computacionales. Adicionalmente, en el caso del conjunto de datos *Multi-Domain Sentiment Analysis*, que originalmente se encontraba en formato *Bag of Words*, las palabras correspondientes a cada documento se unieron en una sola cadena de texto, con el fin de permitir su procesamiento por parte de los modelos de lenguaje.

Para los modelos de tipo *encoder*, se cargaron los *tokenizadores* correspondientes a cada modelo y se aplicó el proceso de *tokenización* sobre el texto limpio. Adicionalmente, las etiquetas de clase se transformaron a formato categórico, y posteriormente se construyeron los *dataloaders* necesarios para el entrenamiento supervisado. A cada modelo se el añadió una cabeza de clasificación (capa densa) y el *fine-tunning* se realizó sobre todos los parámetros del modelo base, buscando ajustar completamente las representaciones al dominio de cada tarea. Cabe resaltar que el conjunto de validación se utilizó para ir evaluando el modelo y guardar aquel con mejor *accuracy*.

Por otro lado, para los modelos de tipo *decoder*, se cargaron los parámetros preentrenados de cada modelo, se *tokenizaron* los datos usando el *tokenizador* correspondiente y se realizó inferencia sobre los textos del conjunto de pruebas mediante el uso de *prompts* o instrucciones específicas diseñadas para cada tarea. Este proceso se repitió iterativamente con diferentes formulaciones de *prompt*, aspecto que se discute con mayor detalle en una sección posterior dedicada al análisis de esta tarea.

3. ANÁLISIS DE DESEMPEÑO

En esta sección se presentan los resultados obtenidos por los distintos modelos en las tareas de clasificación correspondientes a los conjuntos de datos mencionados anteriormente. Para casa caso se reportan las métricas de evaluación calculadas sobre el conjunto de prueba, incluyendo *accuracy, precision, recall* y *F1-score*, con el fin de comparar el rendimiento de las arquitecturas bajo las mismas condiciones experimentales.

- **Resultados del conjunto 20 Newsgroup:** Los modelos *bert-base-uncased* y *distilbert-base-uncased* mostraron un desempeño consistente y elevado. *BERT* alcanzó un accuracy de 0.7954 y un F1-Score Macro de 0.7891, mientras que *DISTILBERT*, a pesar de ser una versión reducida del modelo original, obtuvo resultados comparables con un *accuracy* de 0.7826 y un *F1-score macro* de 0.7748. Estos valores reflejan la capacidad de las arquitecturas *encoder* para capturar relaciones contextuales profundas entre palabras y representar de forma eficiente la estructura semántica de los documentos.

  En contraste, los modelos *decoder* (GPT-2, GPT-2-XL y Gemma2) mostraron un rendimiento significativamente inferior. GPT-2 y GPT-2-XL apenas alcanzaron una *accuracy* de 0.0537, mientras que Gemma2 obtuvo el mejor resultado entre los *decoders* con una *accuracy* de 0.1107 y un *F1-score macro* de 0.0464. Estos valores evidencian las dificultades de los *decoders* para abordar tareas de clasificación con múltiples clases, especialmente cuando no se realiza un entrenamiento supervisado y el desempeño depende exclusivamente de la formulación del *prompt*.

- **Resultados del conjunto Multi-Domain Sentiment Analysis:** En esta tarea binaria de análisis de sentimiento, los modelos encoder volvieron a demostrar un desempeño sobresaliente. DISTILBERT-BASE-UNCASED alcanzó un *accuracy* de 0.8825 y un *F1-score macro* de 0.8825, consolidando su eficacia para la clasificación supervisada incluso en contextos con menor número de clases. Estos resultados reflejan la capacidad de las arquitecturas basadas en *encoders* para generar representaciones semánticas robustas y capturar de manera precisa las diferencias entre opiniones positivas y negativas.

  Por otro lado, los modelos *decoder* presentaron un rendimiento más modesto, aunque superior al observado en el conjunto 20 Newsgroup. GPT-2 y GPT-2-XL alcanzaron *accuracies* de 0.5125 y 0.5500 respectivamente, mientras que Gemma2 obtuvo el mejor desempeño entre los *decoders* con una *accuracy* de 0.6042 y un *F1-score macro* de 0.5533. Si bien los resultados siguen siendo inferiores a los de los modelos *encoder*, se evidencia una mejora relativa atribuible a la menor complejidad de la tarea, en la que los *decoders* pueden apoyarse en patrones de lenguaje más simples y coherentes con los ejemplos de entrada.

En términos generales, los experimentos confirman que los modelos de tipo *encoder* presentan un desempeño superior en tareas de clasificación, especialmente en escenarios multiclase y con estructuras temáticas complejas. Su entrenamiento mediante *fine-tuning* permite ajustar de manera precisa las

representaciones internas al dominio del problema, lo que se traduce en una mayor precisión y estabilidad en las predicciones.

Por otro lado, los modelos *decoder* evidenciaron limitaciones propias de su naturaleza generativa. Su rendimiento depende en gran medida de la formulación del *prompt*, y tienden a producir respuestas inconsistentes cuando los textos a clasificar son complejos o existen múltiples categorías posibles. No obstante, estos modelos demostraron una notable flexibilidad y capacidad de adaptación en tareas más simples, lo que respalda su utilidad en contextos donde no se dispone de un proceso de entrenamiento adicional o se requiere realizar clasificación mediante instrucciones en lenguaje natural.

4. ANÁLISIS DE ROBUSTEZ

El conjunto *20 Newsgroup* (20 clases) representa un escenario mucho más exigente por la combinación de mayor número de categorías y ruido textual (encabezados, citas, fragmentos técnicos). El problema multiclase incrementa la complejidad de las fronteras de decisión y exige representaciones más finas, por lo cual, los *encoders* muestran mayor capacidad de generalización y estabilidad, mientras que los *decoders* presentan dificultades para discriminar tantas clases sin un ajuste supervisado, reflejando menor robustez ante tareas complejas.

En contraste, *Multi-Domain Sentiment Analysis* (clasificación binaria) es una tarea más simple y con señales léxicas más directas, lo que favoreció tanto a *encoders* como a *decoders*. La reducida cantidad de clases facilita la formulación de *prompts* y la interpretación de las salidas generativas, permitiendo que los *decoders* obtengan resultados más competitivos. En resumen, a mayor número de clases y ruido, mejor rendimiento y robustez proporcionan los *encoders* debido a su naturaleza bidireccional, mientras que para tareas más limpias y con un lenguaje más descriptivo, los *decoders* pueden logran un rendimiento aceptable sin la necesidad de re-entrenar el modelo.

5. VENTAJAS Y LIMITACIONES

Las arquitecturas *encoder*, representadas por modelos como *BERT* y *DISTILBERT*, ofrecen ventajas claras para tareas de clasificación supervisada. Su capacidad para generar representaciones contextuales bidireccionales permite capturar relaciones semánticas profundas entre palabras y contextos, lo que se traduce en alta precisión y estabilidad en las predicciones. Además, el proceso de *fine-tuning* posibilita adaptar completamente el modelo al dominio específico del problema. Sin embargo, su principal limitación radica en el costo computacional y de tiempo asociado al entrenamiento, así como en la necesidad de contar con datos etiquetados y recursos de hardware adecuados para realizar el ajuste de parámetros.

Por otro lado, las arquitecturas *decoder*, como *GPT-2*, *GPT-2-XL* y *Gemma2*, destacan por su flexibilidad y facilidad de uso sin requerir entrenamiento adicional. Pueden abordar tareas mediante instrucciones en lenguaje natural, lo que los hace útiles para usuarios con poca experiencia técnica o donde no se desea re-entrenar modelos. No obstante, su desempeño depende en gran medida de la calidad y formulación del *prompt*, y tienden a producir respuestas inconsistentes en tareas complejas o con múltiples clases. Además, al no estar optimizados para clasificación, requieren estrategias adicionales para traducir sus salidas generativas en etiquetas precisas.

6. SELECCIÓN Y SENSIBILIDAD DE LOS PROMPTS

Durante la implementación de los modelos *decoder*, se observó que la formulación del *prompt* tuvo un impacto directo y significativo en el desempeño del modelo. Inicialmente, se utilizaron instrucciones simples sin ejemplos explícitos, lo que produjo respuestas inconsistentes y una alta variabilidad en las predicciones, especialmente en el conjunto 20 Newsgroups, donde la cantidad de clases y la complejidad temática dificultaron la identificación precisa de categorías.

Posteriormente, se diseñaron *prompts* más estructurados que incluían ejemplos de entrada y salida, junto con descripciones detalladas de la tarea. Para el conjunto Multi-Domain Sentiment Analysis, el *prompt* se formuló como una instrucción explícita para clasificar opiniones como *positive* o *negative*, acompañada de ejemplos ilustrativos. En el caso de 20 Newsgroups, se proporcionó el contexto de que los mensajes provenían de foros en línea y se incluyó una lista explícita de categorías disponibles. Este enfoque guiado mejoró notablemente la coherencia y precisión de las respuestas, aunque las limitaciones inherentes de los modelos *decoder* persistieron en tareas con muchas clases. Cabe resaltar que los *prompts* utilizados se pueden encontrar en el *notebook* de *decoders*, específicados de manera explícita en las clases definidas para llevas a cabo el proceso de clasificación.

7. RECOMENDACIONES DE SELECCIÓN

De acuerdo con los resultados obtenidos, se recomienda utilizar modelos *encoder* cuando se cuente con los recursos computacionales y el conocimiento técnico necesarios para realizar *fine-tuning*. Estas arquitecturas resultan especialmente adecuadas para tareas de clasificación con múltiples clases o estructuras semánticas complejas, en las que se requiere capturar relaciones profundas entre los textos y sus categorías. Su entrenamiento supervisado permite ajustar las representaciones internas del modelo al dominio específico, logrando altos niveles de precisión y estabilidad.

Por otro lado, el uso de modelos *decoder* es una alternativa válida en contextos donde no sea posible realizar entrenamiento adicional, no se disponga de datos etiquetados o se busque una solución práctica para tareas de clasificación más simples. Este enfoque es adecuado cuando las categorías son pocas y el lenguaje de entrada no presenta gran complejidad. Sin embargo, para alcanzar un rendimiento comparable al de los *encoders*, que aprovechan el ajuste supervisado, los *decoders* deben contar con una escala considerable, del orden de miles de millones de parámetros, ya que solo así pueden compensar la falta de entrenamiento mediante su capacidad de *aprendizaje en contexto*.