Reporte: Implementación de un Modelo Transformer para Traducción Automática

1. Introducción

En este proyecto, se implementó un modelo Transformer para la tarea de traducción automática del inglés al español. El Transformer es una arquitectura de red neuronal que ha demostrado ser altamente efectiva en tareas de procesamiento de lenguaje natural (NLP), especialmente en traducción automática. El modelo fue entrenado utilizando un conjunto de datos de pares de oraciones en inglés y español, y se evaluó su rendimiento utilizando métricas estándar como BLEU y ROUGE. Asimismo, se visualizaron los **attention_scores** para entender cómo el modelo asigna importancia a diferentes partes de las frases durante la traducción. Además, se exploraron varias técnicas para mejorar el rendimiento del modelo, como el ajuste de hiperparámetros y el uso de embeddings preentrenados.

2. Metodología

1. **Preparación del Entorno y Descarga de Datos:**
    - Se configuró el entorno de trabajo utilizando TensorFlow y Keras.
    - Se descargó y preparó el conjunto de datos de traducción **spa-eng** desde el repositorio de Anki, que contiene pares de oraciones en ambos idiomas.

2. **Preprocesamiento de Datos:**
    - Los datos se parsearon y se dividieron en conjuntos de entrenamiento, validación y prueba.
    - Se aplicó un proceso de vectorización para convertir las oraciones en secuencias de números enteros, lo que es necesario para la entrada del modelo. Se ajustó el tamaño del vocabulario a 20,000 tokens y se limitó la longitud de las secuencias a 20 tokens para el inglés y 21 para el español.

3. **Implementación del Modelo Transformer:**
    - Se implementó un modelo Transformer con las siguientes características:
      - **Encoder**: Capas `MultiHeadAttention` y normalización.
      - **Decoder**: Capas `MultiHeadAttention` , atención sobre el encoder y normalización.

- Se utilizaron embeddings preentrenados de GloVe (Global Vectors for Word Representation) para inicializar los embeddings de palabras en el modelo. Los embeddings de GloVe se descargaron desde el sitio de Stanford NLP y se cargaron en el modelo.
- El modelo fue compilado con un optimizador Adam y una función de pérdida de entropía cruzada categórica. Se implementó un planificador de tasa de aprendizaje exponencial para ajustar dinámicamente la tasa de aprendizaje durante el entrenamiento.

Model: "transformer"

Layer (type)	Output Shape	Param #	Connected to
encoder_inputs (InputLayer)	(None, None)	0	-
decoder_inputs (InputLayer)	(None, None)	0	-
positional_embedding (PositionalEmbedding)	(None, None, 300)	4,506,000	encoder_inputs[0][0]
not_equal (NotEqual)	(None, None)	0	encoder_inputs[0][0]
positional_embedding_1 (PositionalEmbedding)	(None, None, 300)	4,506,000	decoder_inputs[0][0]
transformer_encoder (TransformerEncoder)	[(None, None, 300), (None, 12, None, None)]	5,563,448	positional_embedding[… not_equal[0][0]
not_equal_1 (NotEqual)	(None, None)	0	decoder_inputs[0][0]
transformer_decoder (TransformerDecoder)	[(None, None, 300), (None, 12, None, None), (None, 12, None, None)]	9,895,148	positional_embedding_… transformer_encoder[0… not_equal_1[0][0], not_equal[0][0]
dropout_3 (Dropout)	(None, None, 300)	0	transformer_decoder[0…
dense_4 (Dense)	(None, None, 15000)	4,515,000	dropout_3[0][0]

Total params: 28,985,596 (110.57 MB)
Trainable params: 24,485,596 (93.41 MB)
Non-trainable params: 4,500,000 (17.17 MB)

4. **Entrenamiento del Modelo:**
   - El modelo se entrenó durante 50 épocas, utilizando un planificador de tasa de aprendizaje exponencial.
   - Se implementó una parada temprana (early stopping) para evitar el sobreajuste, deteniendo el entrenamiento si la pérdida de validación no mejoraba después de 3 épocas.

- El modelo entrenado se guardó en disco para su uso posterior, utilizando el formato **.keras**.

5. **Evaluación del Modelo:**
   - Se evaluó el modelo utilizando las métricas BLEU y ROUGE. Estas métricas miden la similitud entre las traducciones generadas por el modelo y las traducciones de referencia.
   - Se generaron traducciones de ejemplo a partir de oraciones de prueba y se compararon con las traducciones de referencia.
   - Se visualizaron los **attention_scores** para entender cómo el modelo asigna importancia a las palabras durante la traducción.
   - Se generaron heatmaps de las activaciones de las capas del modelo, lo que proporcionó insights valiosos sobre el funcionamiento interno del Transformer.

3. Resultados

- **Métricas de Evaluación:**
  - **Precisión en Entrenamiento**: 0.2744
  - **Pérdida en Entrenamiento**: 1.1780
  - **Precisión en Validación**: 0.2403
  - **Pérdida en Validación**: 1.8265

Aunque el modelo ha mejorado, aún existe una brecha entre el rendimiento en entrenamiento y validación, lo que sugiere que el modelo podría estar sobreajustando ligeramente.

- **Traducciones de Ejemplo:**
  - Se generaron traducciones de ejemplo que muestran que el modelo es capaz de capturar el significado general de las oraciones en inglés, aunque a veces comete errores en la elección de palabras o en la estructura gramatical.

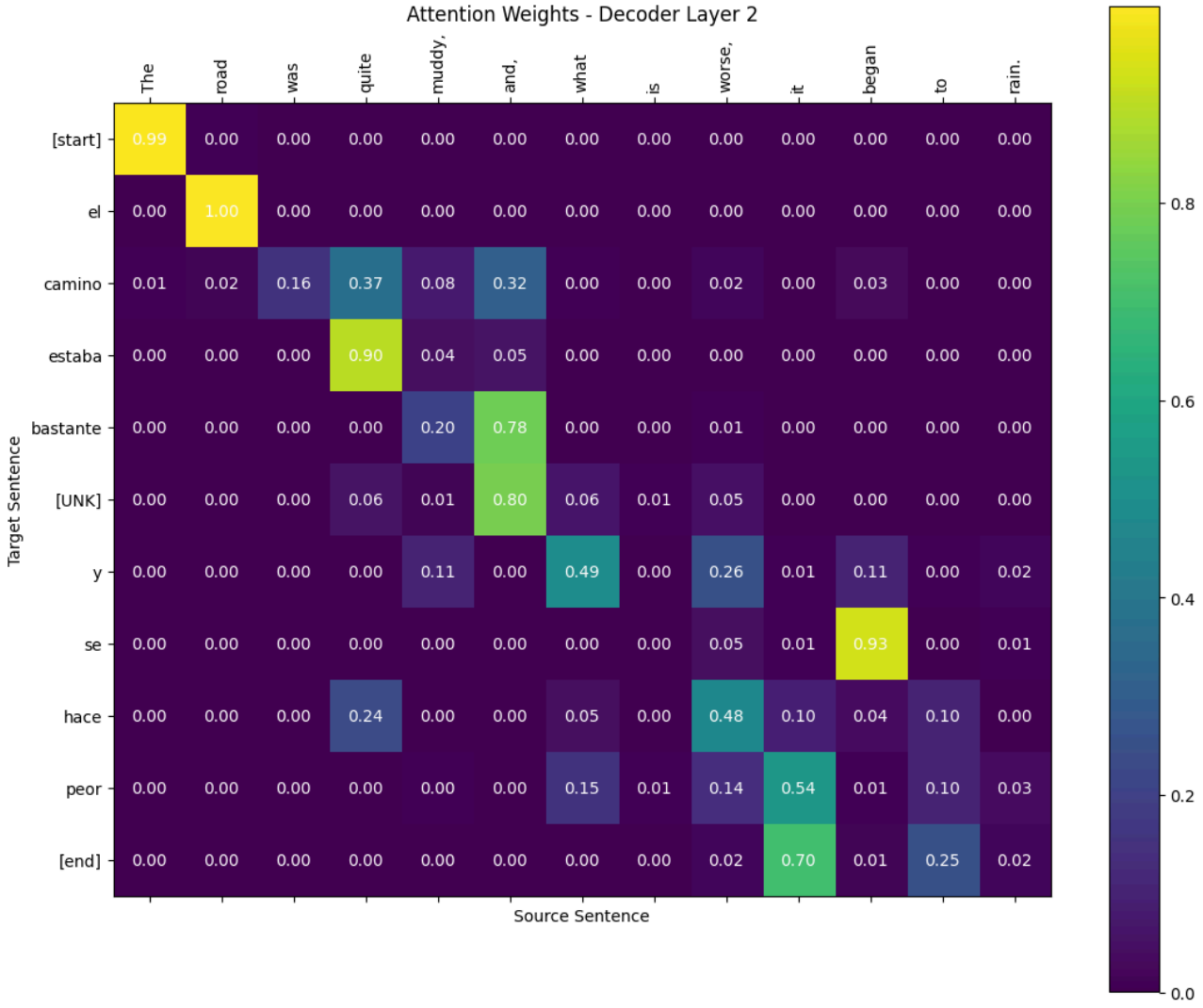
```
Tom introduced Mary to all his friends. [start] tom le me hizo toda su amigo [end]

He really likes baseball. [start] le gusta mucho el béisbol [end]

Tom didn't want to tell Mary the sad news. [start] tom no quería decirle a mary la situación [end]
```

Tom has found Mary. [start] tom ha encontró a mary [end]

Not every book on the desk belongs to me. [start] no se [UNK] cada escritorio a mí [end]

Are you going to take part in the contest? [start] vas a tomar parte en el anillo de final [end]

- **Visualización de Activaciones:**
  - Las visualizaciones de las activaciones de las capas del modelo revelaron que el modelo está aprendiendo a enfocarse en las partes relevantes de las oraciones de entrada para generar las traducciones.
  - Los heatmaps mostraron que las capas de atención del modelo están capturando las relaciones entre las palabras en las oraciones de entrada y salida.



4. Comentarios y Conclusiones

- **Desafíos Encontrados:**
    - Uno de los principales desafíos fue manejar el tamaño del vocabulario y la longitud de las secuencias, lo que requirió ajustes en la vectorización y en la arquitectura del modelo.
    - El entrenamiento del modelo fue computacionalmente costoso, especialmente debido al uso de embeddings preentrenados y a la complejidad del modelo Transformer. Se requirió el uso de una GPU para acelerar el proceso de entrenamiento.
    - **Bajo Rendimiento**: Los scores BLEU y ROUGE son bajos, lo que indica que el modelo no está traduciendo correctamente.
    - **Pesos de Atención**: Los valores altos en el token [start] y la primera palabra en inglés sugieren que el modelo no está aprendiendo a asignar la atención de manera adecuada.

- **Posibles Mejoras:**
    - **Ajuste de Hiperparámetros**: Probar con una tasa de aprendizaje más baja, más épocas o un tamaño de batch más grande.
    - **Arquitectura del Modelo**: Aumentar la dimensión de los embeddings, el número de cabezas de atención o el número de capas.
    - **Regularización**: Añadir dropout o weight decay para evitar el sobreajuste.
    - **Preprocesamiento**: Revisar la tokenización y el manejo de los tokens especiales.
    - Con la disponibilidad de equipo de mejor rendimiento y más tiempo de entrenamiento, podría mejorar considerablemente los resultados

5. Conclusiones

En este proyecto, se implementó un modelo de traducción automática basado en la arquitectura Transformer, integrando embeddings preentrenados de GloVe y evaluando el modelo con métricas BLEU y ROUGE. Aunque el modelo mostró un rendimiento bajo, se identificaron áreas de mejora y se visualizaron los pesos de atención para entender su comportamiento. Futuros trabajos podrían enfocarse en ajustar la arquitectura, los hiperparámetros y el preprocesamiento de los datos para mejorar los resultados.