Documentación del Proyecto: Análisis de Sentimientos en Reseñas de Cine (IMDB)

1. Resumen

Este proyecto aborda el desarrollo de un sistema automatizado para el análisis de sentimientos en reseñas de películas, utilizando técnicas de Procesamiento de Lenguaje Natural (NLP) y Aprendizaje Profundo (Deep Learning). El objetivo principal es clasificar comentarios de texto libre en dos categorías polarizadas: positivo o negativo. Para ello, se implementaron y compararon dos arquitecturas de Redes Neuronales Recurrentes (RNN): *Long Short-Term Memory* (LSTM) y *Gated Recurrent Units* (GRU).

El sistema se entrenó utilizando el conjunto de datos "Large Movie Review Dataset" (IMDB), que consta de 50,000 reseñas balanceadas. El proceso incluyó la vectorización del texto, el entrenamiento supervisado de los modelos y una evaluación exhaustiva mediante métricas estándar como exactitud, precisión, exhaustividad y puntuación F1. Los resultados experimentales mostraron que ambas arquitecturas alcanzan un rendimiento similar, superando el 83% de exactitud en el conjunto de prueba. Finalmente, el modelo con mejor desempeño fue desplegado en una aplicación web interactiva desarrollada con FastAPI, permitiendo a los usuarios finales realizar predicciones de sentimiento en tiempo real sobre nuevas reseñas.

2. Introducción

Motivación

En la era digital actual, el volumen de contenido generado por usuarios en plataformas web ha crecido exponencialmente. Las opiniones de los clientes son activos valiosos para las empresas, estudios de cine y consumidores, ya que influyen directamente en la toma de decisiones y en la reputación de los productos.

Problema

El análisis manual de miles de reseñas es una tarea inviable por su costo temporal y humano. Además, el lenguaje natural es intrínsecamente no estructurado y complejo, presentando desafíos como la ambigüedad, el sarcasmo y la dependencia del contexto, lo que dificulta su clasificación mediante reglas simples o algoritmos tradicionales.

Justificación

El uso de Redes Neuronales Recurrentes (RNNs), específicamente LSTM y GRU, se justifica por su capacidad para procesar datos secuenciales y capturar dependencias a largo plazo en el texto, algo crucial para entender el sentimiento de una oración completa. La automatización de este proceso mediante una interfaz accesible permite democratizar el uso de estas tecnologías avanzadas.

3. Metodología

El desarrollo del sistema siguió un flujo de trabajo de aprendizaje automático estándar, dividido en las siguientes etapas:

1. **Ingesta de Datos:** Carga de archivos de texto crudo desde la estructura de directorios del dataset.
2. **Preprocesamiento y Vectorización:** Limpieza del texto y conversión de palabras a secuencias numéricas utilizando un índice de vocabulario limitado.
3. **Modelado:** Construcción de arquitecturas LSTM y GRU con capas de *Embedding*.
4. **Entrenamiento:** Optimización de los pesos de la red mediante retropropagación (*backpropagation*) y el optimizador Adam.
5. **Evaluación:** Medición del desempeño en datos no vistos.
6. **Despliegue:** Implementación de una API REST y una interfaz gráfica simple.

4. Conjunto de Datos

Se utilizó el **Large Movie Review Dataset (aclImdb)**, un estándar académico para la clasificación de sentimientos binaria.

- **Fuente:** Stanford AI (Maas et al., 2011).
- **Volumen:** 50,000 reseñas en total.
- **Distribución:** Perfectamente balanceado (25,000 positivas, 25,000 negativas).
- **Características:** Las reseñas son textos en inglés de longitud variable.

Ejemplos:

- *Positivo:* "This movie was absolutely fantastic! The acting was superb..."
- *Negativo:* "I was so bored I almost fell asleep. The story went nowhere..."

5. Diseño Experimental

Partición del Conjunto de Datos

Para garantizar la validez de los resultados y evitar el sobreajuste (*overfitting*), los datos se dividieron estrictamente según la estructura original del dataset:

- **Conjunto de Entrenamiento (Train):** 25,000 muestras. Utilizado para ajustar los pesos del modelo. Se aplicó una mezcla aleatoria (*shuffle*) antes de cada época para evitar sesgos por orden.
- **Conjunto de Prueba (Test):** 25,000 muestras. Utilizado para monitorear la pérdida durante el entrenamiento y para la evaluación final de métricas. El modelo nunca "aprendió" de estos datos.

Justificación de Parámetros de Entrenamiento

Se seleccionaron los siguientes hiperparámetros buscando un equilibrio entre rendimiento computacional (entrenamiento en CPU) y capacidad de generalización:

1. **Tamaño del Vocabulario (10,000 palabras):** Se limitó a las 10,000 palabras más frecuentes. Según la Ley de Zipf, la mayoría de las palabras son raras y aportan poco valor semántico general ("ruido"), mientras que aumentan drásticamente la dimensionalidad y el tiempo de cómputo.
2. **Longitud Máxima de Secuencia (150 tokens):** Se truncaron o rellenaron las reseñas a 150 palabras. El sentimiento suele establecerse al inicio o final del texto; secuencias más largas aumentan el costo computacional sin una ganancia significativa en precisión para esta tarea.
3. **Dimensiones del Embedding (64):** Un vector de 64 dimensiones es suficiente para capturar relaciones semánticas en un vocabulario de 10k palabras sin crear un modelo excesivamente pesado.
4. **Unidades RNN (32 neuronas):** Se usaron 32 unidades en las capas LSTM/GRU. Un número mayor (ej. 128) tiende a memorizar los datos de entrenamiento (*overfitting*) en datasets pequeños/medianos y ralentiza el entrenamiento. 32 unidades fuerzan al modelo a aprender patrones generales.
5. **Épocas (3):** Se observó empíricamente que el modelo converge rápidamente. Entrenar por más épocas resultaba en sobreajuste, donde la precisión en entrenamiento subía pero en prueba bajaba.

Medidas de Desempeño

Para evaluar la calidad del clasificador se utilizaron:

- **Exactitud (Accuracy):** Porcentaje global de aciertos.
- **Precisión (Precision):** Calidad de los positivos detectados (cuántos de los que predijo positivos realmente lo eran).
- **Exhaustividad (Recall):** Cantidad de positivos reales que el modelo fue capaz de encontrar.
- **F1-Score:** Media armónica entre precisión y exhaustividad.
- **Matriz de Confusión:** Visualización de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

6. Resultados

Tras el entrenamiento y la evaluación, se obtuvieron los siguientes resultados en el conjunto de prueba:

Métrica	Valor Obtenido
Exactitud (Accuracy)	**83.09%**
Precisión	79.89%
Exhaustividad (Recall)	88.46%
F1-Score	83.95%

Análisis: El modelo LSTM alcanzó una exactitud del **83.09%**, demostrando ser robusto para la tarea. La exhaustividad (Recall) es particularmente alta (88%), lo que indica que el modelo es muy bueno detectando reseñas positivas, aunque a veces clasifica erróneamente algunas negativas como positivas (falsos positivos), como se evidenció en casos de sarcasmo o reseñas mixtas.

Ambas arquitecturas (LSTM y GRU) mostraron un rendimiento muy similar, con diferencias menores al 1%, lo cual es consistente con la literatura para tareas de complejidad media como esta.