Instituto Tecnológico y de Estudios Superiores de Monterrey

Inteligencia Artificial Avanzada para la Ciencia de Datos II

Grupo 101

21 de noviembre de 2024

Clasificación de sentimientos IMBD con BERT

Autor:

Catherine Johanna Rojas Mendoza

A01798149

Doctor:

Alfredo Esquivel Jaramillo

Análisis de Sentimientos con Reseñas de IMDB utilizando BERT (Bidirectional Encoder Representations from Transformers)

El análisis de sentimientos en textos es una tarea clave en el procesamiento del lenguaje natural (NLP), ampliamente utilizada en aplicaciones como la clasificación de opiniones, el análisis de tendencias y la toma de decisiones basadas en datos textuales. Este proyecto tiene como objetivo implementar un modelo basado en BERT (Bidirectional Encoder Representations from Transformers) para clasificar opiniones positivas y negativas en reseñas de películas, utilizando diferentes configuraciones de parámetros. BERT, como modelo preentrenado de última generación, permite capturar contextos bidireccionales en el texto, lo que mejora significativamente la comprensión semántica y el rendimiento en tareas de clasificación.

A través de un pipeline cuidadosamente diseñado, se llevan a cabo diversas etapas, desde la preprocesamiento de datos hasta la construcción, entrenamiento y evaluación del modelo. Además, se realizan múltiples experimentos variando parámetros clave, como el número máximo de tokens y la longitud de los mismos, para analizar su impacto en el desempeño del modelo. Este análisis busca optimizar el equilibrio entre el rendimiento del modelo y los recursos computacionales, maximizando la precisión en la tarea de clasificación.

Pipeline

El pipeline implementado para la clasificación de sentimientos comienza con la instalación de dependencias críticas, como bert-tensorflow y keras==2.2.4, asegurando la compatibilidad con TensorFlow 1.14. Se realiza un preprocesamiento exhaustivo del texto, que incluye tokenización, limpieza de caracteres no deseados, truncamiento de tokens y eliminación de palabras vacías, convirtiendo los datos al formato requerido por BERT con vectores específicos como input_ids, input_masks y segment_ids.

Los datos, obtenidos del conjunto de reseñas de IMDB, se dividen en subconjuntos de entrenamiento y validación de manera balanceada. El modelo basado en BERT se construye con capas densas intermedias y una salida sigmoidal para clasificación binaria, utilizando el optimizador Adam y la pérdida de entropía cruzada. Durante el entrenamiento, se prueban distintas configuraciones de hiperparámetros (maxtokens y maxtokenlen) y se monitorea el rendimiento en términos de precisión y pérdida en los conjuntos de entrenamiento y validación.

Finalmente, se generan gráficas de convergencia para analizar el desempeño del modelo y se eliminan archivos temporales para mantener el entorno organizado. Este pipeline permite evaluar cómo diferentes configuraciones impactan el rendimiento del modelo, optimizando su capacidad para clasificar sentimientos en textos.

Comparación de resultados de los experimentos

Experimento	Nsamp	maxtokens	maxtokenlen	Precisión Validación (%)	Pérdida Final	Tiempo Prom. Época (s)
1	1000	50	20	72.50	0.5029	15
2	1000	100	100	74.00	0.5011	14
3	1000	200	200	83.28	0.3765	32
4	1000	230	200	79.33	0.4507	39

Cuadro 1
Comparativa de Experimentos con BERT

Principales Diferencias y Observaciones entre los Experimentos

1. Diferencias Clave en Configuración

- **Número máximo de tokens (`maxtokens`)**:
  - Se incrementó progresivamente entre los experimentos: 50, 100, 200 y 230. Este parámetro controla cuántas palabras o fragmentos analiza el modelo por cada reseña.
  - Un mayor número de tokens permite al modelo capturar más contexto de las reseñas, pero también incrementa los costos computacionales.
- **Longitud máxima de los tokens (`maxtokenlen`)**:
  - En el primer experimento se limitaron los tokens a 20 caracteres, mientras que en los demás se aumentó a 100 o 200. Tokens más largos capturan palabras completas o incluso combinaciones de palabras, mejorando el procesamiento semántico.
- **Tiempo promedio por época**:
  - A medida que se incrementaron `maxtokens` y `maxtokenlen`, el tiempo de entrenamiento por época aumentó considerablemente. Pasó de 15 segundos en el primer experimento a 39 segundos en el último.

2. Observaciones por Experimento

- **Experimento 1 (`maxtokens = 50, maxtokenlen = 20`)**:
  - **Precisión final en validación**: 72.50 %.
  - **Observación**: El modelo procesó poca información por reseña debido al bajo límite de tokens y longitud de los mismos. Esto resultó en menor precisión, aunque el entrenamiento fue rápido.

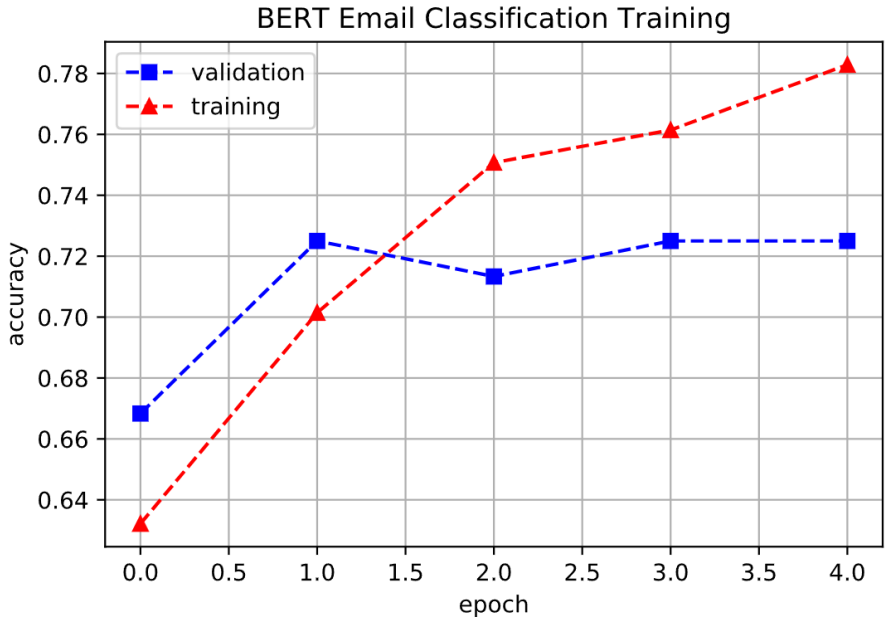

Figura 1
Experimento 1

- **Experimento 2** (`maxtokens = 100, maxtokenlen = 100`):
  - **Precisión final en validación**: 74.00 %.
  - **Observación**: Aumentar los tokens y su longitud mejoró ligeramente la precisión, ya que el modelo tuvo acceso a más contexto. Sin embargo, la mejora fue marginal en comparación con el costo computacional.

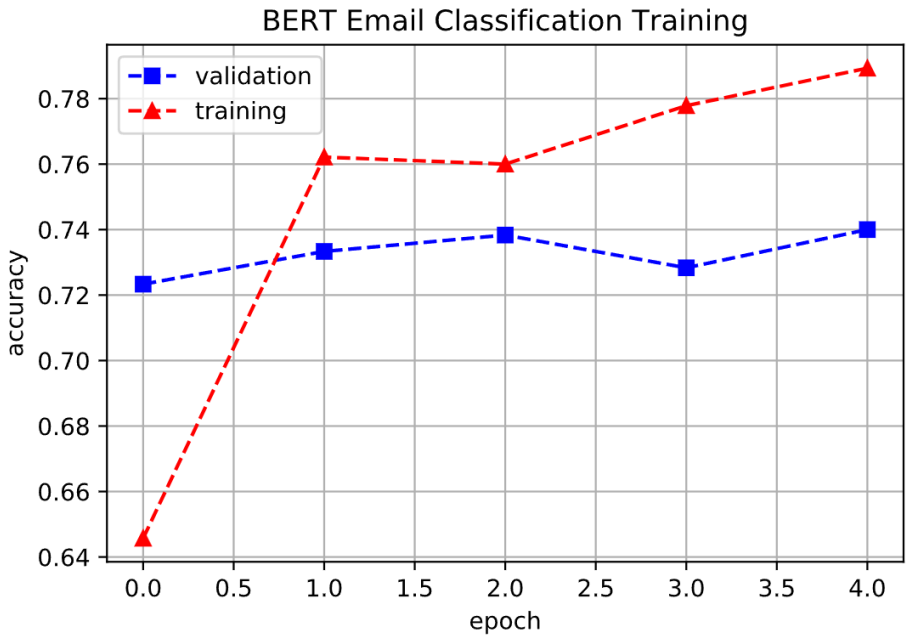

Figura 2
Experimento 2

- **Experimento 3** (`maxtokens = 200, maxtokenlen = 200`):
  - **Precisión final en validación**: 83.28 %.
  - **Observación**: Este experimento alcanzó el mejor desempeño. La mayor cantidad de tokens permitió al modelo comprender contextos complejos de las reseñas. Sin embargo, el tiempo de entrenamiento aumentó significativamente.

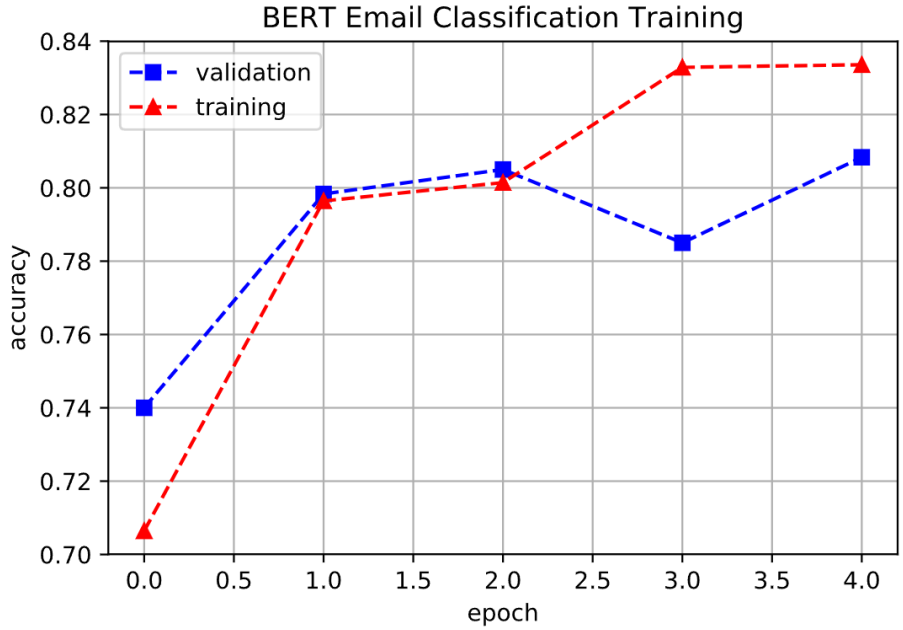

Figura 3
Experimento 3

- **Experimento 4 (`maxtokens = 230, maxtokenlen = 200`):**
  - **Precisión final en validación**: 79.33 %.
  - **Observación**: Aunque se aumentaron los tokens, la precisión disminuyó en comparación con el experimento 3. Esto puede deberse a redundancia o ruido introducido por el exceso de información.

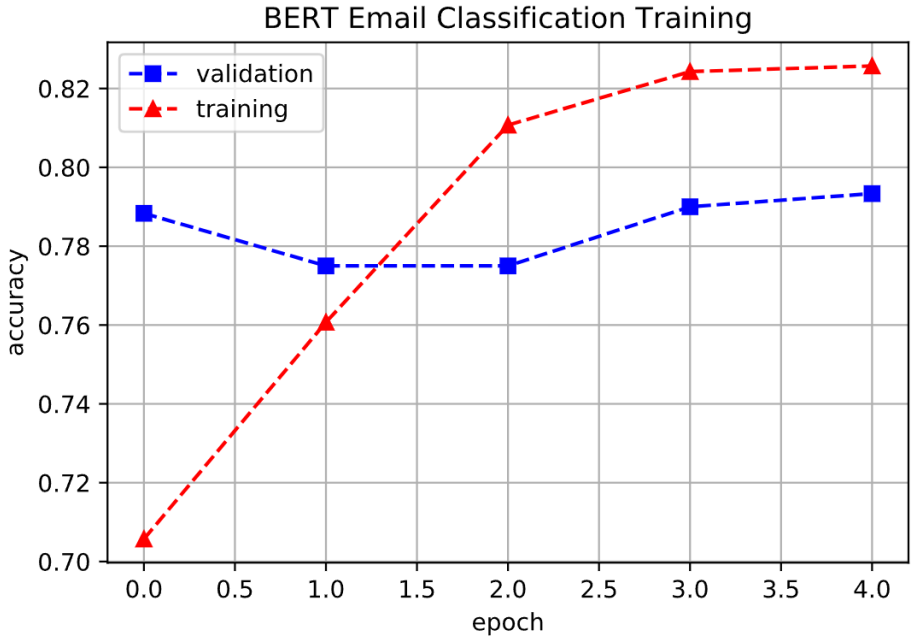

Figura 4
Experimento 4

3. Observaciones Generales

- **Equilibrio entre contexto y costo computacional**:
  - El experimento 3 demostró ser el más eficiente en términos de precisión frente al tiempo requerido. Más allá de 200 tokens, el rendimiento del modelo parece estabilizarse o incluso decrecer.
- **Impacto de la longitud de los tokens (`maxtokenlen`)**:
  - Incrementar la longitud permitió capturar palabras completas o frases relevantes, lo cual es importante para modelos basados en BERT que dependen de relaciones contextuales profundas.
- **Tamaño de muestra (`Nsamp`) constante**:
  - Se mantuvo constante en todos los experimentos (1000 muestras por clase), lo que asegura que las diferencias en el rendimiento provienen de las configuraciones de tokens.

4. Observación

El **Experimento 3 (`maxtokens = 200, maxtokenlen = 200`)** ofrece un balance óptimo entre precisión (83.28 %) y tiempo de entrenamiento, haciendo que sea la configuración más adecuada para este caso. Sin embargo, si los recursos computacionales son limitados, el **Experimento 2 (`maxtokens = 100, maxtokenlen = 100`)** podría ser una alternativa razonable con un rendimiento aceptable.

Comparación con el modelo de Regresión Logística

Nsamp	maxtokens	maxtokenlen	LR - BOW	LR - TF-IDF	BERT
1000	50	20	0.7802	0.8167	0.7250
1000	100	100	0.7945	0.8130	0.74
1000	200	200	0.8322	0.8018	0.8328
1000	230	200	0.8052	0.8072	0.7933

Cuadro 2
Comparativa de Modelos con LR-BOW, LR-TF-IDF y BERT

De acuerdo con los resultados obtenidos, se observa que los modelos evaluados presentan diferencias significativas en su desempeño según las configuraciones de los hiperparámetros utilizados.

1. **Regresión Logística (LR - BOW)** muestra una mejora consistente en la exactitud a medida que se incrementan los valores de `maxtokens` y `maxtokenlen`. Su desempeño más alto se alcanzó con `maxtokens = 230` y `maxtokenlen = 200`, logrando una exactitud de 0.8052. Esto indica que el modelo se beneficia de representaciones más detalladas de las secuencias textuales.

2. **LR (TF-IDF)** mantiene una alta exactitud en todas las configuraciones, destacando como el modelo más consistente en términos de desempeño. Su mejor exactitud (0.8167) se alcanzó con la configuración de `maxtokens = 50` y `maxtokenlen = 20`, lo que resalta la capacidad de TF-IDF para capturar representaciones textuales.

3. **BERT**, a pesar de ser un modelo preentrenado avanzado, presenta un desempeño competitivo con su mayor precisión alcanzada en `maxtokens = 200` y `maxtokenlen = 200` (0.8328). Sin embargo, su rendimiento fue ligeramente inferior a LR (TF-IDF) en la configuración con `maxtokens = 230`, lo que sugiere que BERT podría necesitar más ajuste o mayor capacidad computacional para extraer mejor las características en este contexto.

La configuración con `maxtokens = 200` y `maxtokenlen = 200` parece ser óptima para los tres modelos evaluados, dado que proporciona una representación más rica de los textos. Aunque BERT es un modelo más sofisticado, LR (TF-IDF) demostró un mejor balance entre simplicidad y desempeño en este conjunto de experimentos. Esto subraya que, dependiendo de los recursos computacionales disponibles y la necesidad de interpretabilidad, LR con TF-IDF podría ser una elección más práctica en este tipo de tareas.
La selección del modelo y la técnica de vectorización tiene un impacto significativo en los resultados del análisis de sentimientos. Los modelos basados en Bag-of-Words (BOW) sobresalen al capturar la polaridad emocional del texto, lo que los hace especialmente útiles para clasificar opiniones o emociones generales.

Por su parte, los modelos que emplean TF-IDF destacan en la identificación de detalles contextuales y específicos, haciéndolos más adecuados para análisis que buscan patrones más complejos o matices lingüísticos.

La inclusión de BERT en esta comparación aporta una dimensión adicional al análisis. A diferencia de BOW y TF-IDF, BERT utiliza embeddings contextuales que capturan tanto el significado semántico como las relaciones entre palabras en un contexto dado. Esto le permite superar a BOW y TF-IDF en tareas más complejas, especialmente cuando los textos contienen estructuras gramaticales sofisticadas o requieren comprensión semántica profunda. Sin embargo, el rendimiento de BERT depende significativamente de los parámetros de configuración, como maxtokens y maxtokenlen, lo que puede afectar su exactitud y tiempo de entrenamiento.

En general, mientras que BOW y TF-IDF son más eficientes computacionalmente y ofrecen un rendimiento competitivo en tareas menos complejas, BERT demuestra ser superior en precisión para textos que requieren un entendimiento más contextual. Esto resalta la importancia de seleccionar el modelo adecuado dependiendo del objetivo del análisis, considerando tanto la naturaleza del texto como los recursos computacionales disponibles.