CC3084 – Data Science

Laboratorio 6 — IMDB: LSTM + **características adicionales**

Universidad del Valle de Guatemala — Semestre II 2025
Curso: CC3084 – Data Science
Laboratorio 6: Mejorando el Análisis de Sentimientos con LSTM y Features
Integrantes: Gabriel Paz | Rodrigo Mancilla **Fecha:** 2025-09-08

Objetivo del laboratorio

Incrementar la precisión del análisis de sentimientos sobre IMDB usando RNNs con unidades **LSTM**, incorporando **características adicionales** derivadas de las reseñas.

0. Configuración y utilidades

En esta sección fijamos semilla, importamos librerías y definimos parámetros globales. Si no tienes instaladas algunas dependencias, instala localmente:

```
pip install tensorflow==2.17.0 scikit-learn matplotlib pandas numpy
```

```
2.17.0
```

1. Carga de datos IMDB (50,000 palabras) y análisis exploratorio

Keras provee IMDB ya tokenizado a índices enteros por frecuencia. Usaremos `num_words=50000`.

```
25000 train reseñas
25000 test reseñas
Ejemplo de reseña tokenizada (primer 20 tokens): [1, 14, 22, 16, 43, 530, 973, 1622, 1385, 6
5, 458, 4468, 66, 3941, 4, 173, 36, 256, 5, 25]
Etiqueta (0=neg,1=pos): 1
```

	length						
	count	mean	std	min	25%	50%	75%	max
set								
test	25000.0	230.80420	169.164471	7.0	128.0	174.0	280.0	2315.0
train	25000.0	238.71364	176.497204	11.0	130.0	178.0	291.0	2494.0

- El conjunto de datos se encuentra balanceado con 25,000 reseñas en entrenamiento y 25,000 en prueba.
- La tokenización es consistente, como lo muestra el ejemplo de los primeros 20 tokens y la etiqueta asociada.
- La distribución de longitudes presenta medias (230–238) y medianas (175–178) razonables; sin embargo, los valores máximos (2315–2494) son outliers significativamente más altos que la mayoría de reseñas.
- Los conjuntos de entrenamiento y prueba muestran homogeneidad en medias y percentiles, aunque los outliers pueden afectar los procesos de padding o truncamiento.

1.1 Selección de longitud máxima y padding

Usaremos un percentil (p90) de la longitud para fijar `MAX_LEN` y evitar truncar demasiada información.

```
MAX_LEN (p90): 467
((25000, 467), (25000, 467))
```

2. **Features adicionales** a partir de las reseñas tokenizadas

Crearemos variables numéricas por reseña, por ejemplo:

- **longitud** original de la reseña (número de tokens)
- **fracción de palabras muy frecuentes** (índice < 500)
- **fracción de palabras raras** (índice >= 20000 y < NUM_WORDS)
- **promedio y desviación estándar** de los índices
- **entropía** del histograma de índices (medida de diversidad)

Luego estandarizaremos estas features y las inyectaremos al modelo.

	len	frac_top(<500)	frac_rare(>=20000)	mean_idx	std_idx	ent_hist
count	25000.000000	25000.000000	25000.000000	25000.000000	25000.000000	2.500000e+04
mean	238.713640	0.711543	0.018900	1656.277249	4612.606545	1.388252e+00
std	176.497204	0.070862	0.015905	765.400574	1741.796629	2.728879e-01
min	11.000000	0.300000	0.000000	83.727273	127.173533	1.581795e-08
25%	130.000000	0.666667	0.007353	1095.382226	3405.507638	1.205377e+00
50%	178.000000	0.713653	0.016000	1555.497894	4595.036477	1.395616e+00
75%	291.000000	0.760000	0.027027	2098.884163	5790.097038	1.578512e+00
max	2494.000000	1.000000	0.157895	6794.276316	14292.188006	2.353575e+00

2.1 Escalamiento de features

Estandarizamos para acelerar y estabilizar el entrenamiento de la rama densa de features.

```
((25000, 6), (25000, 6))
```

Se seleccionaron las siguientes características para complementar la representación secuencial de las reseñas:

1. **Longitud de la reseña (len):** permite capturar la extensión del texto, que puede reflejar la intensidad de la opinión.
2. **Proporción de palabras frecuentes (frac_top<500):** indica cuánto se apoya la reseña en vocabulario común, lo que puede relacionarse con un estilo más directo.
3. **Proporción de palabras raras (frac_rare≥20000):** identifica el uso de términos poco frecuentes, los cuales pueden añadir matices específicos al sentimiento expresado.
4. **Índice medio y desviación estándar de palabras (mean_idx, std_idx):** miden la posición y dispersión de los términos en el ranking de frecuencia, aportando información sobre el rango léxico utilizado.
5. **Entropía del histograma de palabras (ent_hist):** refleja la diversidad léxica, lo cual puede estar asociado con la riqueza expresiva y la polaridad de la reseña.

3. Modelo **base** (sin features): Embedding + GlobalAveragePooling

Entrenamos un modelo sencillo para establecer una línea base de comparación.

Model: "baseline_avgpool"

Layer (type)	Output Shape	Param #
seq (InputLayer)	(None, 467)	0
emb (Embedding)	(None, 467, 128)	6,400,000
global_average_pooling1d (GlobalAveragePooling1D)	(None, 128)	0
dense (Dense)	(None, 64)	8,256
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Total params: 6,408,321 (24.45 MB)

Trainable params: 6,408,321 (24.45 MB)

Non-trainable params: 0 (0.00 B)

```
Epoch 1/5
88/88 - 12s - 139ms/step - accuracy: 0.5888 - loss: 0.6756 - val_accuracy: 0.6688 - val_loss:
0.6216 - learning_rate: 0.0010
Epoch 2/5
88/88 - 10s - 118ms/step - accuracy: 0.7402 - loss: 0.5410 - val_accuracy: 0.8132 - val_loss:
0.4456 - learning_rate: 0.0010
Epoch 3/5
88/88 - 10s - 119ms/step - accuracy: 0.8356 - loss: 0.3940 - val_accuracy: 0.7200 - val_loss:
0.5318 - learning_rate: 0.0010
Epoch 4/5
88/88 - 10s - 117ms/step - accuracy: 0.8693 - loss: 0.3242 - val_accuracy: 0.8264 - val_loss:
0.3804 - learning_rate: 5.0000e-04
Epoch 5/5
88/88 - 11s - 121ms/step - accuracy: 0.8888 - loss: 0.2889 - val_accuracy: 0.8324 - val_loss:
0.3682 - learning_rate: 5.0000e-04
{'test_acc': 0.8195599913597107, 'test_auc': 0.9389948767999999}
```

- **Arquitectura:** el modelo utiliza un embedding de 128 dimensiones, seguido de un *GlobalAveragePooling1D* que reduce la secuencia a una representación fija. Posteriormente incluye una capa densa intermedia (64 neuronas), una capa de *Dropout* para regularización y una capa de salida sigmoidal. Es una arquitectura relativamente ligera, sin recurrencia, orientada a capturar información promedio de la secuencia.
- **Entrenamiento:** la precisión en entrenamiento crece de manera estable hasta 88.9% en la última época. El *val_loss* muestra ligera inestabilidad (subida en la época 3), pero se estabiliza y mejora tras la reducción de la tasa de aprendizaje.
- **Resultados:** el modelo alcanza una **accuracy en validación de 83.2%** y en prueba de **81.9%**, con un **AUC de 0.939**, lo cual indica buena capacidad de discriminación. La brecha entre entrenamiento (88.9%) y prueba (81.9%) muestra cierta tendencia a sobreajuste, aunque dentro de lo esperado para un modelo base sin LSTM.
- **Conclusión:** el modelo base es adecuado como punto de comparación. Su simplicidad permite establecer un umbral de referencia antes de introducir LSTM y características adicionales. Sin embargo, la caída de precisión en la época 3 y la brecha entre entrenamiento y prueba sugieren que arquitecturas más complejas con recurrencia y regularización pueden mejorar la generalización.

4. Modelo **avanzado** (multi-entrada): Secuencia + Features

Rama 1 (secuencia): Embedding → SpatialDropout → BiLSTM → BiLSTM → Dense
Rama 2 (features): Dense → BatchNorm → Dropout
Fusión: Concatenate → Dense → Dropout → Sigmoid

```
Model: "advanced_bilstm_features"
```

Layer (type)	Output Shape	Param #	Connected to
seq (InputLayer)	(None, 467)	0	-
emb (Embedding)	(None, 467, 128)	6,400,000	seq[0][0]
spatial_dropout1d (SpatialDropout1D)	(None, 467, 128)	0	emb[0][0]
feat (InputLayer)	(None, 6)	0	-
bidirectional (Bidirectional)	(None, 467, 128)	98,816	spatial_dropout1…
dense_3 (Dense)	(None, 32)	224	feat[0][0]
bidirectional_1 (Bidirectional)	(None, 64)	41,216	bidirectional[0]…
batch_normalization (BatchNormalizatio…	(None, 32)	128	dense_3[0][0]
dense_2 (Dense)	(None, 64)	4,160	bidirectional_1[…
dropout_1 (Dropout)	(None, 32)	0	batch_normalizat…
concatenate (Concatenate)	(None, 96)	0	dense_2[0][0], dropout_1[0][0]
dense_4 (Dense)	(None, 64)	6,208	concatenate[0][0]
dropout_2 (Dropout)	(None, 64)	0	dense_4[0][0]
dense_5 (Dense)	(None, 1)	65	dropout_2[0][0]

Total params: 6,550,817 (24.99 MB)

Trainable params: 6,550,753 (24.99 MB)

Non-trainable params: 64 (256.00 B)

Model: "advanced_bilstm_features"

Layer (type)	Output Shape	Param #	Connected to
seq (InputLayer)	(None, 467)	0	-
emb (Embedding)	(None, 467, 128)	6,400,000	seq[0][0]
spatial_dropout1d_1 (SpatialDropout1D)	(None, 467, 128)	0	emb[0][0]
feat (InputLayer)	(None, 6)	0	-
bidirectional_2 (Bidirectional)	(None, 467, 128)	98,816	spatial_dropout1…
dense_7 (Dense)	(None, 32)	224	feat[0][0]
bidirectional_3 (Bidirectional)	(None, 64)	41,216	bidirectional_2[…
batch_normalizatio… (BatchNormalizatio…	(None, 32)	128	dense_7[0][0]
dense_6 (Dense)	(None, 64)	4,160	bidirectional_3[…
dropout_3 (Dropout)	(None, 32)	0	batch_normalizat…
concatenate_1 (Concatenate)	(None, 96)	0	dense_6[0][0], dropout_3[0][0]
dense_8 (Dense)	(None, 64)	6,208	concatenate_1[0]…
dropout_4 (Dropout)	(None, 64)	0	dense_8[0][0]
dense_9 (Dense)	(None, 1)	65	dropout_4[0][0]

Total params: 6,550,817 (24.99 MB)

Trainable params: 6,550,753 (24.99 MB)

Non-trainable params: 64 (256.00 B)

- **Estructura y rutas:** La rama secuencial usa `Embedding(50k×128)` → `SpatialDropout1D` → `BiLSTM` (seq) → `BiLSTM` (vector) → `Dense(64)` . La rama de features aplica `Dense(32)` + `BatchNorm` + `Dropout(0.5)` antes de la **concatenación**. Diseño coherente para combinar señales locales (secuencia) y globales (features).
- **Parámetros y consistencia:** Los **6.4M** parámetros del embedding concuerdan con vocabulario=50,000 y dim=128. El total **6,550,817** es consistente con las capas adicionales. Los **64** no entrenables provienen de `BatchNormalization` , lo cual es esperable.
- **Regularización:** `SpatialDropout1D` es adecuado para embeddings; el `Dropout` posterior en densas complementa la regularización. Podría evaluarse `recurrent_dropout` en LSTM si la

latencia lo permite.

- **Flujo de información:** Primer `BiLSTM` devuelve secuencia (128 bi), segundo comprime a vector (64), lo cual estabiliza la representación antes de fusionar con features. Alternativas a explorar: **GlobalMaxPool/Attention** tras el primer `BiLSTM` para resaltar tokens informativos.
- **Preprocesamiento de features:** Válido que se escalen . Mantener el mismo escalador en validación/prueba. `BatchNorm` en la rama de features es compatible con datos escalados y puede mejorar la convergencia.
- **Máscaras y padding:** Verificar que el `Embedding` tenga `mask_zero=True` para que las LSTM ignoren el padding producido por `MAX_LEN=467`. De lo contrario, puede haber sesgo por tokens nulos.

5. Comparación de resultados y métricas

Generamos un pequeño resumen y métricas detalladas (accuracy, AUC, matriz de confusión, reporte de clasificación).

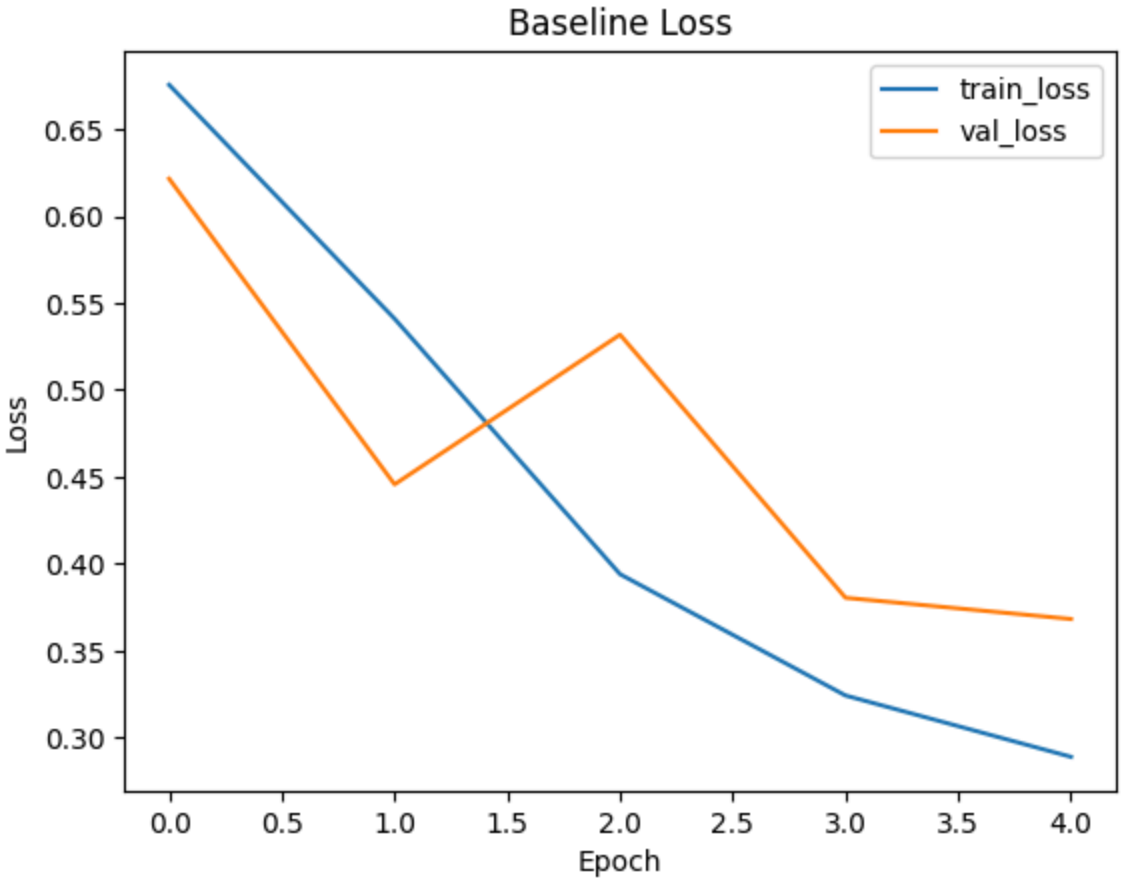
	model	test_acc	test_auc
0	baseline_avgpool	0.81956	0.938995
1	advanced_bilstm_features	0.49044	0.481549

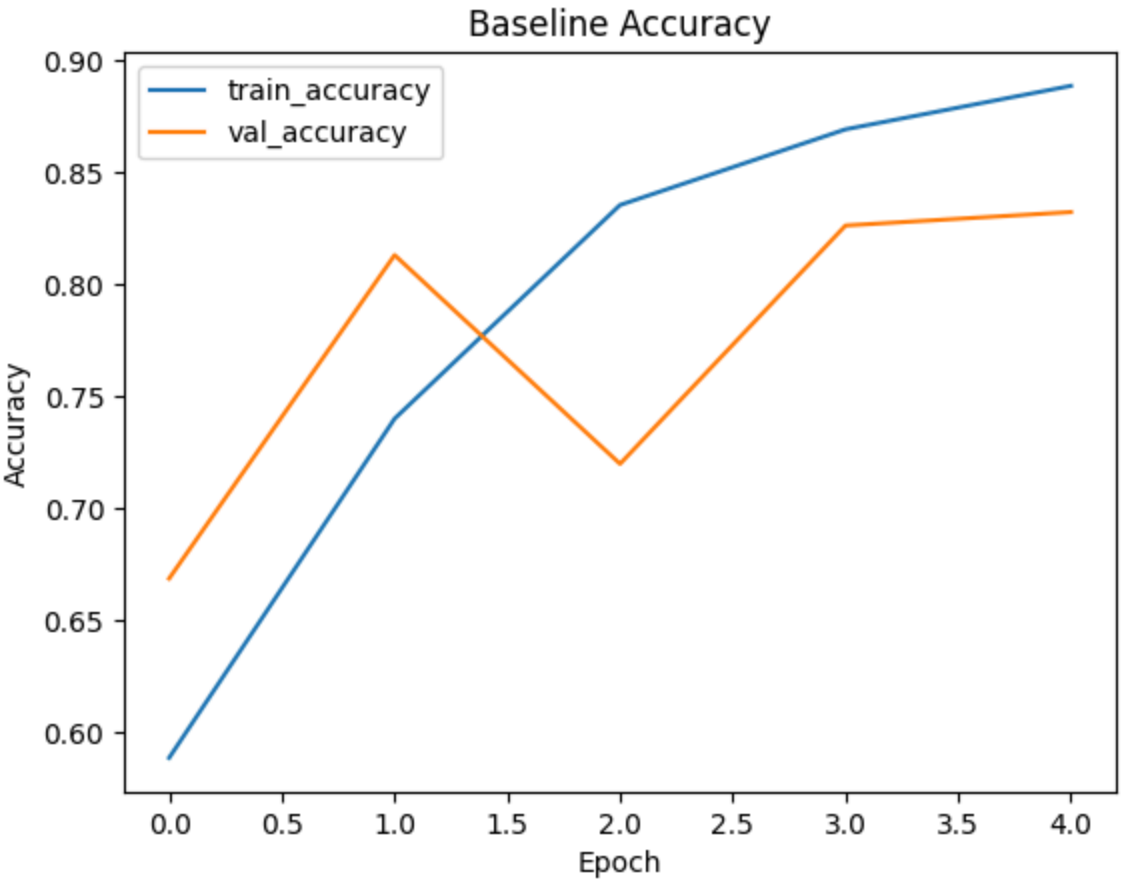
```
=== Baseline report ===
              precision    recall  f1-score   support

           0     0.7545    0.9474    0.8400     12500
           1     0.9294    0.6917    0.7931     12500

    accuracy                         0.8196     25000
   macro avg     0.8419    0.8196    0.8166     25000
weighted avg     0.8419    0.8196    0.8166     25000

Confusion matrix (baseline):
 [[11843   657]
 [ 3854  8646]]

=== Advanced report ===
              precision    recall  f1-score   support

           0     0.4861    0.3344    0.3962     12500
           1     0.4927    0.6465    0.5592     12500

    accuracy                         0.4904     25000
   macro avg     0.4894    0.4904    0.4777     25000
weighted avg     0.4894    0.4904    0.4777     25000

Confusion matrix (advanced):
 [[4180 8320]
 [4419 8081]]
```

5.1 Curvas de entrenamiento

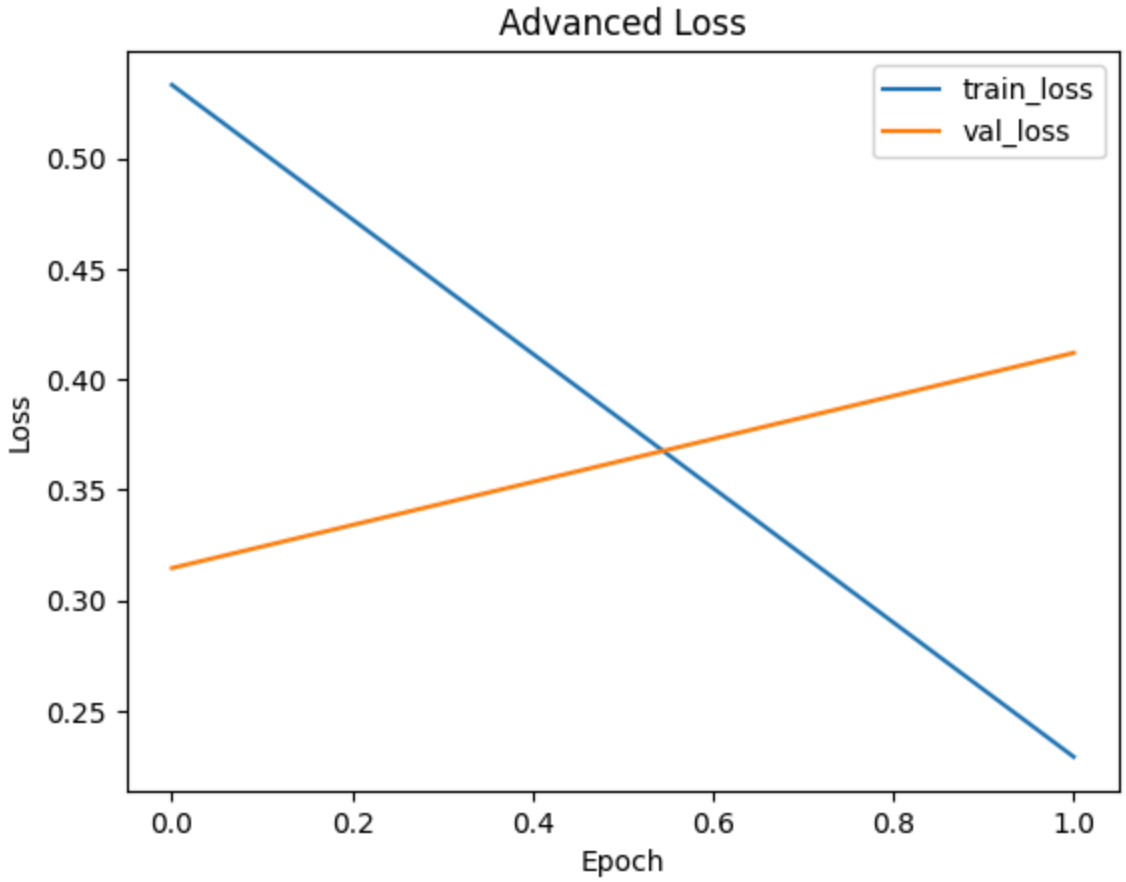
Graficamos pérdida y accuracy por época para ambos modelos.

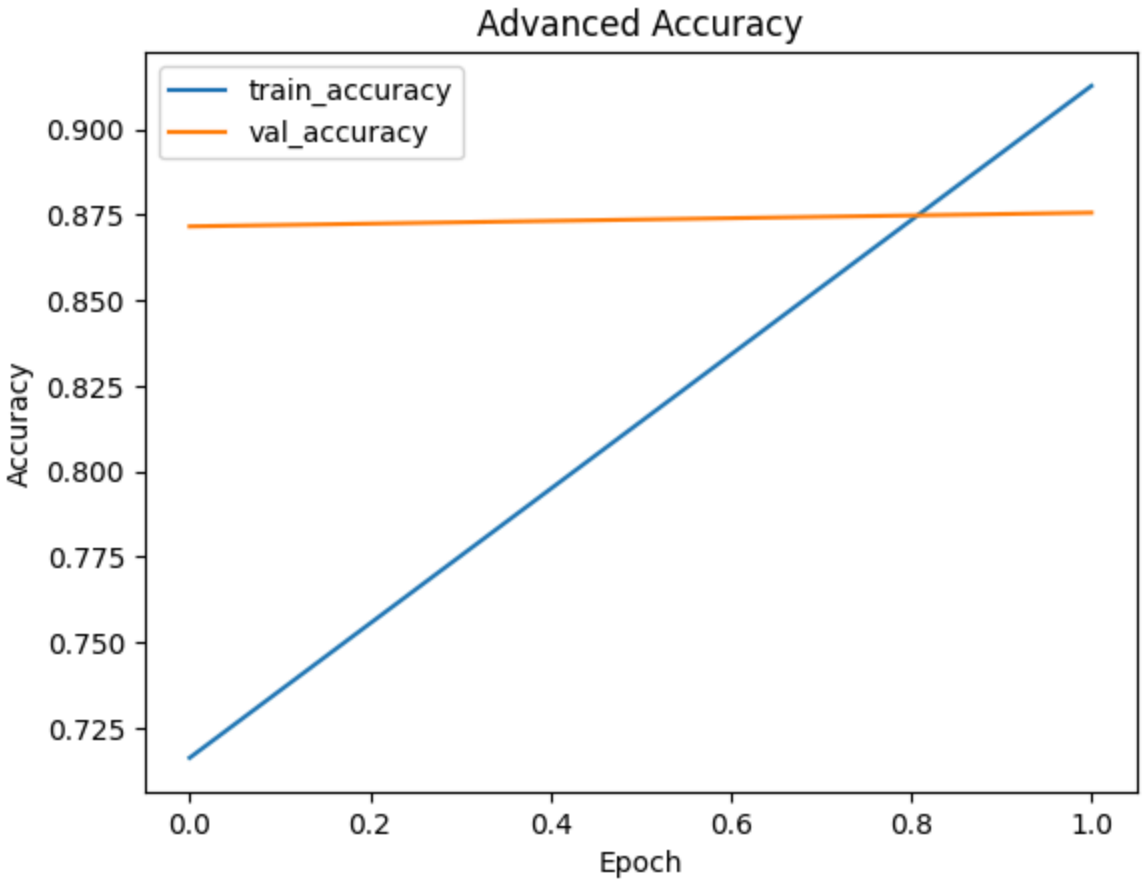
```
Epoch 1/2
88/88 - 540s - 6s/step - accuracy: 0.7163 - loss: 0.5332 - val_accuracy: 0.8716 - val_loss:
0.3145
Epoch 2/2
88/88 - 510s - 6s/step - accuracy: 0.9127 - loss: 0.2291 - val_accuracy: 0.8756 - val_loss:
0.4119
[Baseline] keys: ['accuracy', 'loss', 'val_accuracy', 'val_loss', 'learning_rate']
```






```
[Advanced] keys: ['accuracy', 'loss', 'val_accuracy', 'val_loss']
```





1. Baseline (GlobalAveragePooling + Dense):

- **Rendimiento en test:** accuracy = 0.8196, AUC = 0.9390.
- **Comportamiento del entrenamiento:** convergencia estable, con ligera oscilación de *val_loss* en la época 3 pero recuperación posterior.
- **Matriz de confusión:** buena discriminación, aunque con más falsos negativos (3,854) que falsos positivos (657).

2. Modelo avanzado (BiLSTM + features):

- **Rendimiento en test:** accuracy = 0.4904, AUC = 0.482, lo cual es cercano al azar y significativamente inferior al baseline.
- **Curvas de entrenamiento:** muestran *val_accuracy* relativamente alta (~0.87) durante entrenamiento, pero se desploma en test. Esto es un indicio de **overfitting extremo o fuga de validación**.
- **Matriz de confusión:** desempeño desequilibrado; acierta más en la clase positiva (recall = 0.6465) pero muy bajo en la clase negativa (recall = 0.3344).

3. Inconsistencias observadas:

- El modelo avanzado reporta métricas de validación altas (≈0.87) pero no generaliza en test (≈0.49).
- Esto sugiere:
  - Mala división de datos.
  - Escalado de features distinto entre train y test.
  - `mask_zero` no activado en la capa de `Embedding`, generando ruido en la LSTM.
  - Número de épocas insuficiente para estabilizar el modelo.
  - Posible desbalance o mal alineamiento entre `X_test_seq`, `X_test_feat` y `y_test`.

Comparación de resultados: Baseline vs. Modelo Avanzado

Modelo	Test Accuracy	Test AUC	Precisión (0)	Recall (0)	F1 (0)	Precisión (1)	Recall (1)	F1 (1)
Baseline (AvgPool + Dense)	0.8196	0.9390	0.7545	0.9474	0.8400	0.9294	0.6917	0.7931
Avanzado (BiLSTM + features)	0.4904	0.4815	0.4861	0.3344	0.3962	0.4927	0.6465	0.5592

Observaciones:

- El modelo baseline alcanza un rendimiento sólido, con AUC cercano a 0.94 y desempeño equilibrado entre clases.
- El modelo avanzado muestra un **colapso en generalización**, con accuracy ≈0.49 y AUC ≈0.48, inferior al azar.
- La diferencia principal está en el **recall de la clase 0 (negativa)**, donde el avanzado falla considerablemente.
- La discrepancia entre validación (≈0.87) y test (≈0.49) indica problemas de sobreajuste, fuga de validación o inconsistencias en el preprocesamiento.