

# Reporte

---

Autora: Selene Martínez Ventura

## 1, 2. Ejecución y preentrenamiento del modelo desde cero.

1. Adaptación de *English-to-Spanish NMT by fchollet*

[https://keras.io/examples/nlp/neural\\_machine\\_translation\\_with\\_transformer/](https://keras.io/examples/nlp/neural_machine_translation_with_transformer/)

2. Se descargo el conjunto de datos de Anki *English-to-Spanish*

3. Ejecución:

- Se implementa el modelo transformador de secuencia a secuencia
- Entrenamiento para realizar traducciones.
- Se realizo el entrenamiento desde cero con 100 epocas.
- Se almacena el modelo preentrenado para el uso futuro.
- Se obtuvieron resultados ROUGE

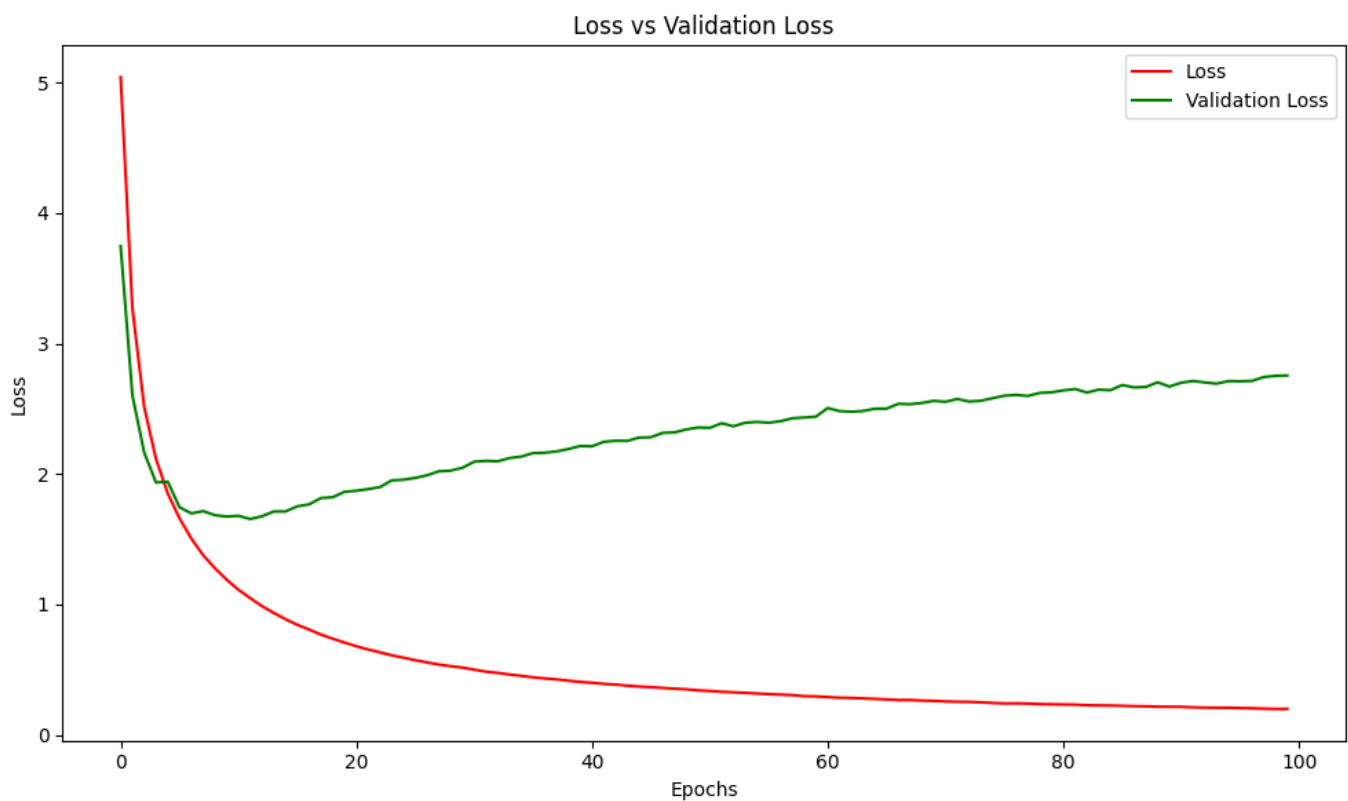
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

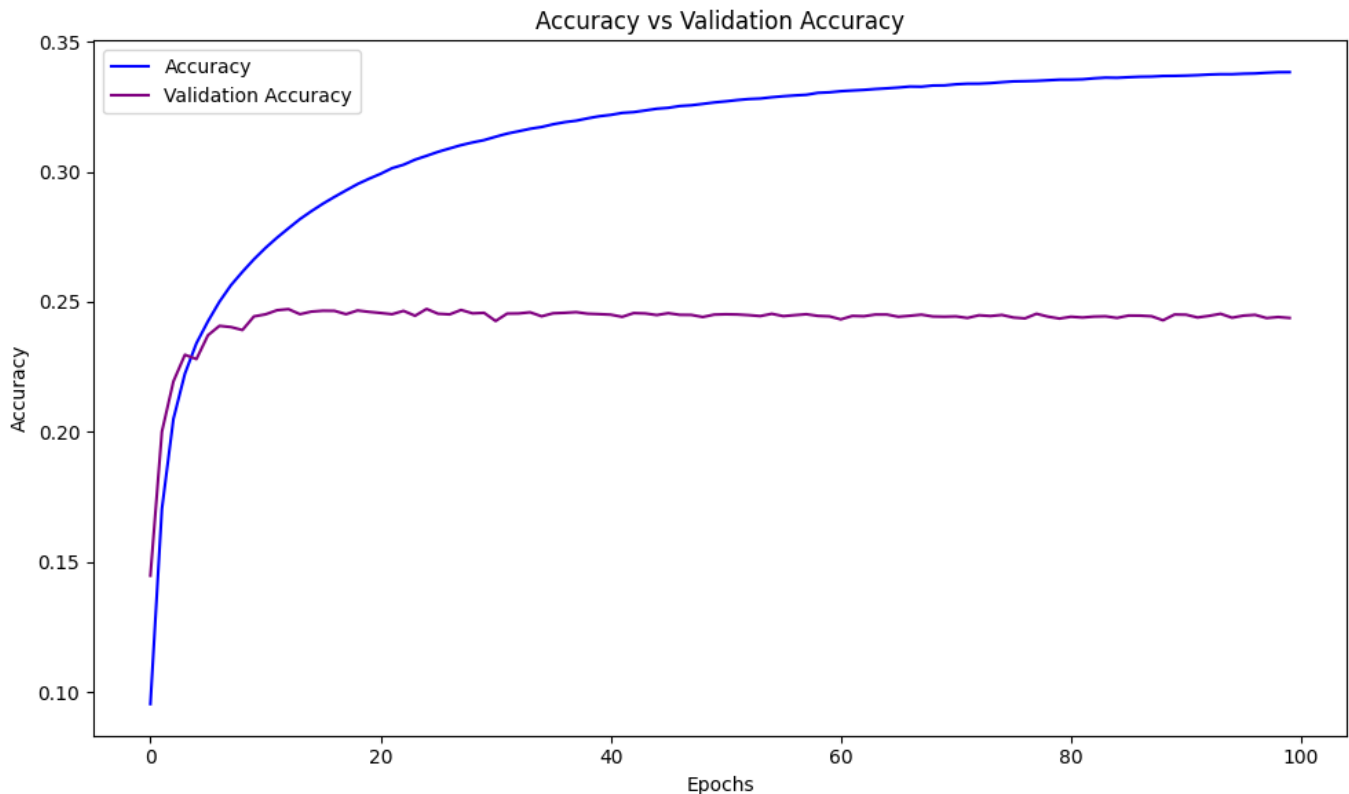
```
#df_results = pd.read_csv("/content/drive/MyDrive/training_logs.csv",
usecols=["Accuracy", "loss"]).head(100)
df_results = pd.read_csv("training_logs.csv").head(100) #, usecols=
["accuracy", "loss"]).head(100)
losses = df_results["loss"].values
accuracies = df_results["accuracy"].values
val_accuracies = df_results["val_accuracy"].values
val_losses = df_results["val_loss"].values
```

```
import matplotlib.pyplot as plt

# Crear la figura para Loss y Validation Loss
plt.figure(figsize=(10, 6))
plt.plot(losses, label='Loss', color='red')
plt.plot(val_losses, label='Validation Loss', color='green')
plt.title('Loss vs Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.tight_layout()
plt.show()
```

```
# Crear la figura para Accuracy y Validation Accuracy
plt.figure(figsize=(10, 6))
plt.plot(accuracies, label='Accuracy', color='blue')
plt.plot(val_accuracies, label='Validation Accuracy', color='purple')
plt.title('Accuracy vs Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.tight_layout()
plt.show()
```





A pesar del entrenamiento de 100 épocas la precisión se quedó estancada entre el 30% y el 35% al rededor de la época 60. Esto con la métrica "accuracy".

#### **val\_accuracy:**

A medida que avanzan las épocas, la precisión de validación muestra una tendencia ascendente, aunque de forma más lenta que la precisión en el conjunto de entrenamiento. Esto sugiere que el modelo está mejorando gradualmente en generalizar a los datos de validación.

#### **val\_loss:**

La pérdida de validación sigue una tendencia descendente, aunque con cierta irregularidad. Inicialmente es bastante alta y luego disminuye. En algunos puntos parece estabilizarse o aumentar ligeramente, esto puede indicar que el modelo comienza a ajustarse más específicamente a los datos de entrenamiento, al rededor de la epoca 20. Sin embargo, el aumento no es fuerte, por lo cual el sobreajuste no es severo sin embargo es algo que podria mejorar realizando cambios en los parámetros del modelo como la tasa de aprendizaje, el número de ngrams, Early Stopping, entre otros.

#### **accuracy:**

La tendencia es positiva, hasta al rededor del 0.34 en la última epoca, lo cual indica la mejora del modelo para clasificar correctamente los ejemplos de entrenamiento a medida que avanza el entrenamiento.

#### **loss:**

La tendencia es negativa. Lo cual muestra que el modelo está reduciendo el error en el conjunto de entrenamiento. En general, una pérdida más baja indica que el modelo está haciendo mejores predicciones

en los datos de entrenamiento lo cual se refleja a partir de la epoca 30. Esta tendencia a partir de la epoca 30 indica que el modelo está entrenando adecuadamente.

```
df_rouge = pd.read_csv("rouge_scores.csv")

# Resumen de las estadísticas descriptivas
print("Estadísticas descriptivas del DataFrame:")
print(df_rouge.describe())
```

Estadísticas descriptivas del DataFrame:

	ROUGE-1 Precision	ROUGE-1 Recall	ROUGE-1 F1-score	ROUGE-2
Precision	ROUGE-2 Recall	ROUGE-2 F1-score		
count	300.000000	300.000000	300.000000	
300.000000	300.000000	300.000000		
mean	0.754754	0.749635	0.747616	
0.512039	0.508723	0.507142		
std	0.183819	0.182834	0.176205	
0.287661	0.288509	0.285305		
min	0.181818	0.200000	0.190476	
0.000000	0.000000	0.000000		
25%	0.625000	0.625000	0.615385	
0.285714	0.300000	0.293651		
50%	0.773504	0.750000	0.761905	
0.500000	0.500000	0.500000		
75%	0.878472	0.875000	0.875000	
0.714286	0.700000	0.666667		
max	1.000000	1.000000	1.000000	
1.000000	1.000000	1.000000		

```
pd.set_option("display.max_colwidth",50)
pd.set_option("display.max_columns", None) # Mostrar todas las columnas
pd.set_option("display.width", 130) # Ajustar el ancho de la tabla
print(df_rouge.head(10)) # Imprimir las primeras 10 filas
```

	Input	
Reference \		
0	This article is misleading.	[start] El artículo es confuso. [end]
1	I know you've been trying to help.	[start] Sé que has estado intentando ayudar. [...
2	She told him to stop.	[start] Ella le dijo que parase. [end]
3	I encouraged Tom to learn French.	[start] Alenté a Tom a aprender francés. [end]
4	The foreign minister attended the talks.	[start] El ministro de Exteriores asistió a la...

5	Are you afraid of Tom?	[start]	¿Te
	da miedo Tom? [end]		
6	I have family in Los Angeles.	[start]	Tengo parientes en
	Los Ángeles. [end]		
7	You're the only one who can help me.	[start]	Eres el único que
	puede ayudarme. [end]		
8	I'm actually quite happy with that.	[start]	Lo cierto es que estoy
	bastante satisf...		
9	We're watching.	[start]	
	Estamos mirando. [end]		
Translated ROUGE-1 Precision			
ROUGE-1 Recall	ROUGE-1 F1-score	ROUGE-2 Precision \	
0	[start] el artículo es [UNK] [end]		0.857143
0.857143	0.857143	0.666667	
1	[start] sé que has estado tratando de ayudar [...		0.777778
0.875000	0.823529	0.625000	
2	[start] ella le dijo que parara [end]		0.857143
0.857143	0.857143	0.666667	
3	[start] [UNK] a tom le inglés con tom [end]		0.500000
0.555556	0.526316	0.111111	
4	[start] el ministro de música [UNK] [end]		0.625000
0.500000	0.555556	0.428571	
5	[start] le tienes miedo a tom [end]		0.571429
0.666667	0.615385	0.166667	
6	[start] tengo familia en los Ángeles de chicos...		0.666667
0.857143	0.750000	0.375000	
7	[start] usted es la única que puede ayudarme [...		0.555556
0.625000	0.588235	0.375000	
8	[start] la verdad es que estoy bastante conten...		0.636364
0.636364	0.636364	0.300000	
9	[start] estamos observando [end]		0.750000
0.750000	0.750000	0.333333	
ROUGE-2 Recall ROUGE-2 F1-score			
0	0.666667	0.666667	
1	0.714286	0.666667	
2	0.666667	0.666667	
3	0.125000	0.117647	
4	0.333333	0.375000	
5	0.200000	0.181818	
6	0.500000	0.428571	
7	0.428571	0.400000	
8	0.300000	0.300000	
9	0.333333	0.333333	

### 3. Guardar el modelo

Se almaceno el modelo completo preentrenado en *English\_to\_spanish.keras*

### Carga del modelo preentrenado

## 4. Embeddings

Fuente: [https://keras.io/examples/nlp/pretrained\\_word\\_embeddings/](https://keras.io/examples/nlp/pretrained_word_embeddings/)

- Se descargaron los 'embeddings' de GloVe (822M zip file).
- El archivo contiene vectores codificados con texto de varios tamaños: 50, 100, 200 y 300 dimensiones. En este ejercicio se usó el de 100 dimensiones *glove.6B.100d.txt*

## 5. Layer activations

- Tutorial de referencia para la generación de las capas de activación  
<https://github.com/tensorflow/text/blob/master/docs/tutorials/transformer.ipynb>

Se pretendió visualizar los pesos de atención sin éxito. Nota: Personalmente requiero más tiempo para estudiar lo que necesito adaptar para la correcta ejecución de este código. Hice intentos para adaptar el código y definir la clase que utiliza `return_attention_scores=True` para posteriormente obtener los `attention_weights` del modelo transformer.

## 6, 7. Mejorando el performance.

### BLEU y ROUGE

Se realizaron las pruebas con los siguientes parámetros:

#### Caso 1.

- ngrams = 2
- optimizador = adam
- learning rate 1e-3
- épocas 50
- Se guardaron los resultados del entrenamiento en *training\_logs\_1.csv*
- Se guardaron los scores de ROUGE y BLEU en los archivos *rouge\_scores\_1 bleu\_scores\_1*

#### Se descarta por pérdida de datos ROUGE y BLEU

#### Caso 2.

- ngrams = 2
- optimizador = RMSprop
- learning rate 1e-3, 5e-4 patience=8
- épocas 50, se detiene en época 26, 14 épocas
- Training completed in 178.71 seconds
- Se guardaron los resultados del entrenamiento en *training\_logs\_2.csv*
- Se guardaron los scores de ROUGE y BLEU en los archivos *rouge\_scores\_2 bleu\_scores\_2*

#### Caso 3.

- ngrams = 5

- optimizador = adam
- learning rate 5e-4
- epocas 50
- Training completed in 187.54
- Se guardaron los resultados del entrenamiento en *training\_logs\_3.csv*
- Se guardaron los scores de ROUGE y BLEU en los archivos *rouge\_scores\_3* *bleu\_scores\_3*

Caso 4.

- ngrams = 5
- optimizador = RMSprop
- learning rate 1e-4
- epocas 50
- Se guardaron los resultados del entrenamiento en *training\_logs\_4.csv*
- Se guardaron los scores de ROUGE y BLEU en los archivos *rouge\_scores\_4* *bleu\_scores\_4*

Caso 5.

- ngrams = 7
- optimizador = RMSprop
- learning rate 1e-4
- epocas 50
- Se guardaron los resultados del entrenamiento en *training\_logs\_7.csv*
- Se guardaron los scores de ROUGE y BLEU en los archivos *rouge\_scores\_7* *bleu\_scores\_7*

Resultados

Training results

```
df_results_1 = pd.read_csv("training_logs_1.csv") # adam, , 1e-3, ngram = 2
df_results_2 = pd.read_csv("training_logs_2.csv") # RMSprop , 1e-3, ngram = 2
df_results_3 = pd.read_csv("training_logs_3.csv") # adam, 5e-4, ngram = 5
df_results_4 = pd.read_csv("training_logs_4.csv") # RMSprop, 1e-4, ngram = 5
df_results_7 = pd.read_csv("training_logs_7.csv") # RMSprop, 1e-4, ngram = 7

# Resumen de las estadísticas descriptivas
print("Estadísticas descriptivas del DataFrame:")
print(df_results_7.describe())
```

Estadísticas descriptivas del DataFrame:					
	epoch	Accuracy	loss	val_Accuracy	val_loss
count	50.00000	50.000000	50.000000	50.000000	50.000000
mean	24.50000	0.701202	1.565244	0.706700	1.466865
std	14.57738	0.081562	0.767314	0.063385	0.536045

min	0.00000	0.431505	0.874022	0.504951	1.002744
25%	12.25000	0.659950	1.049277	0.677256	1.092972
50%	24.50000	0.723799	1.321729	0.728620	1.275681
75%	36.75000	0.764382	1.794111	0.756399	1.632197
max	49.00000	0.791968	4.819674	0.770724	3.507289

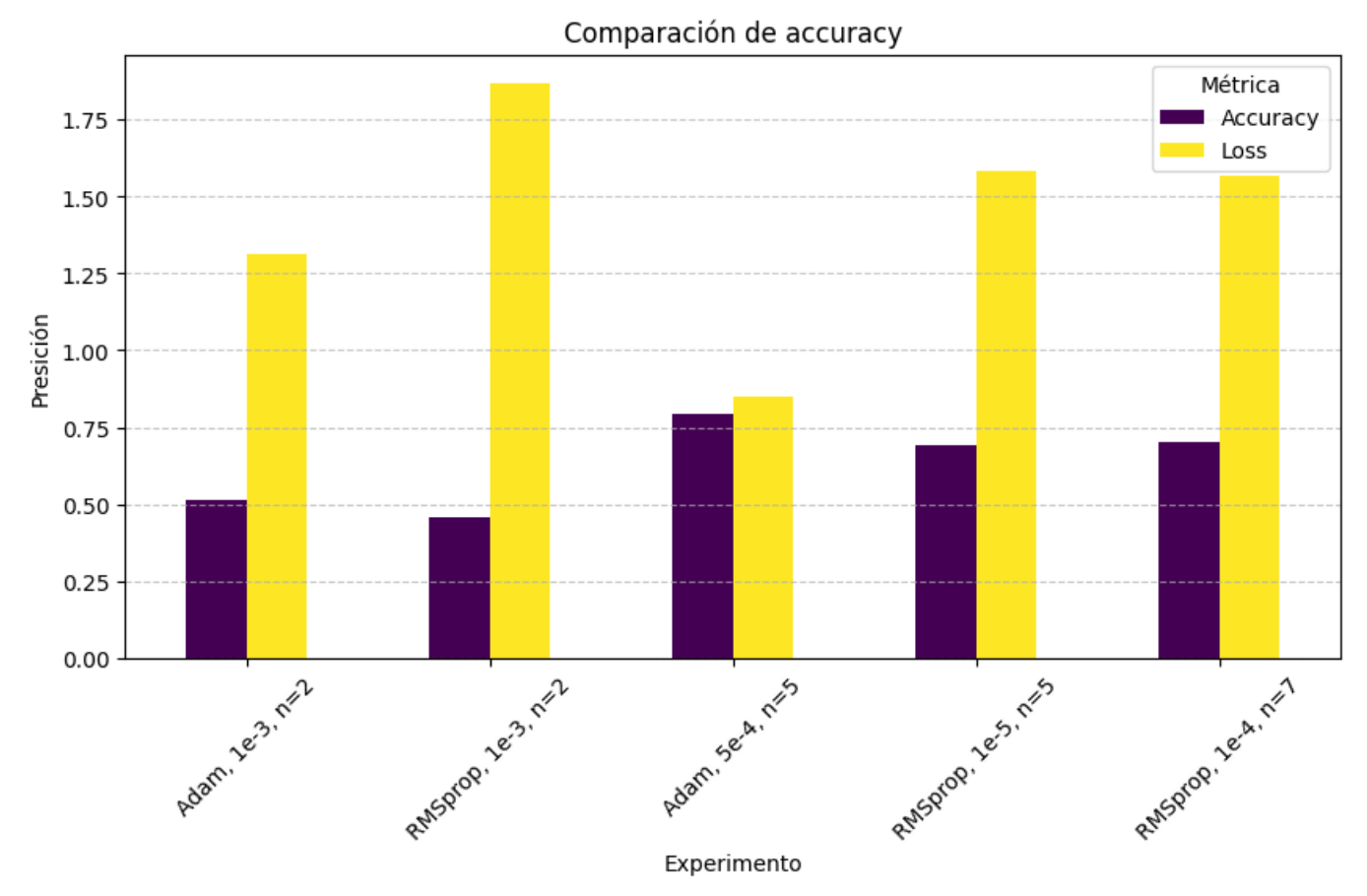
```
# Crear un DataFrame resumen con valores promedio
summary_data = {
    "Experimento": ["Adam, 1e-3, n=2", "RMSprop, 1e-3, n=2", "Adam, 5e-4, n=5", "RMSprop, 1e-5, n=5", "RMSprop, 1e-4, n=7",],
    "Accuracy": [
        df_results_1["Accuracy"].mean(),
        df_results_2["Accuracy"].mean(),
        df_results_3["Accuracy"].mean(),
        df_results_4["Accuracy"].mean(),
        df_results_7["Accuracy"].mean(),
    ],
    "Loss": [
        df_results_1["loss"].mean(),
        df_results_2["loss"].mean(),
        df_results_3["loss"].mean(),
        df_results_4["loss"].mean(),
        df_results_7["loss"].mean(),
    ],
}

df_summary = pd.DataFrame(summary_data)
print(df_summary)
```

	Experimento	Accuracy	Loss
0	Adam, 1e-3, n=2	0.513656	1.314256
1	RMSprop, 1e-3, n=2	0.458405	1.865882
2	Adam, 5e-4, n=5	0.791448	0.849425
3	RMSprop, 1e-5, n=5	0.692802	1.580144
4	RMSprop, 1e-4, n=7	0.701202	1.565244

```
fig, ax = plt.subplots(figsize=(10, 5))
df_summary.set_index("Experimento").plot(kind="bar", ax=ax, cmap="viridis")
plt.title("Comparación de accuracy")
plt.ylabel("Presición")
plt.xticks(rotation=45)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.legend(title="Métrica")
plt.show()
```





ROUGE

ROUGE-1(1-grama) Precision: Mide la precisión de los 1-gramas (palabras individuales) que se generan en la traducción. Ejemplo si la precisión de ROUGE-1 es 0.625, estov significa que el 62.5% de las palabras en la traducción generada coinciden con las palabras en la traducción de referencia.

ROUGE-1 Recall: Mide el recall de los 1-gramas (palabras individuales), es decir, cuántas palabras de la referencia están presentes en la traducción generada. Ejemplo: si el recall de ROUGE-1 es 0.8333, esto significa que el modelo generó el 83.33% de las palabras que están en la traducción de referencia.

ROUGE-1 F1-score: Es la media armónica entre la precisión y el recall. Ejemplo: Si F1-score de ROUGE-1 es 0.7143, esto sugiere que el modelo tiene un buen equilibrio entre precisión y recall en las palabras individuales.

ROUGE-2(2-grama) Precision: Mide la precisión de los 2-gramas generados. Ejemplo: La precisión de ROUGE-2 es 0.2857, lo que indica que solo el 28.57% de los pares de palabras en la traducción generada coinciden con la referencia.

ROUGE-2 Recall: Mide el recall de los 2-gramas, es decir, cuántos pares de palabras de la referencia están presentes en la traducción generada. Ejemplo: El recall de ROUGE-2 es 0.4, lo que indica que el modelo generó el 40% de los pares de palabras que estaban en la traducción de referencia.

ROUGE-2 F1-score: Es la media armónica entre la precisión y el recall para los 2-gramas. Ejemplo: El F1-score de ROUGE-2 es 0.3333, lo que sugiere que el modelo tiene una precisión y un recall más bajos para los pares de palabras.

```
#df_rouge_1 = pd.read_csv("rouge_scores_1.csv") # adam, , 1e-3, ngram = 2
df_rouge_2 = pd.read_csv("rouge_scores_2.csv") # RMSprop , 1e-3, ngram = 2
df_rouge_3 = pd.read_csv("rouge_scores_3.csv") # adam, 5e-4, ngram = 5
df_rouge_4 = pd.read_csv("rouge_scores_4.csv") # RMSprop, 1e-5, ngram = 5
df_rouge_7 = pd.read_csv("rouge_scores_7.csv")

# Resumen de las estadísticas descriptivas
print("Estadísticas descriptivas del DataFrame:")
print(df_rouge_7.describe())
```

Estadísticas descriptivas del DataFrame:

	ROUGE-1 Precision	ROUGE-1 Recall	ROUGE-1 F1-score	ROUGE-2
Precision	ROUGE-2 Recall	ROUGE-2 F1-score		
count	100.000000	100.000000	100.000000	
100.000000	100.000000	100.000000		
mean	0.718549	0.696986	0.703606	
0.428131	0.420770	0.421978		
std	0.161483	0.173471	0.161751	
0.287618	0.288944	0.286778		
min	0.428571	0.375000	0.400000	
0.000000	0.000000	0.000000		
25%	0.600000	0.571429	0.571429	
0.222222	0.191667	0.200000		
50%	0.714286	0.700000	0.705882	
0.387500	0.400000	0.387500		
75%	0.839286	0.800000	0.804545	
0.600000	0.571429	0.575630		
max	1.000000	1.000000	1.000000	
1.000000	1.000000	1.000000		

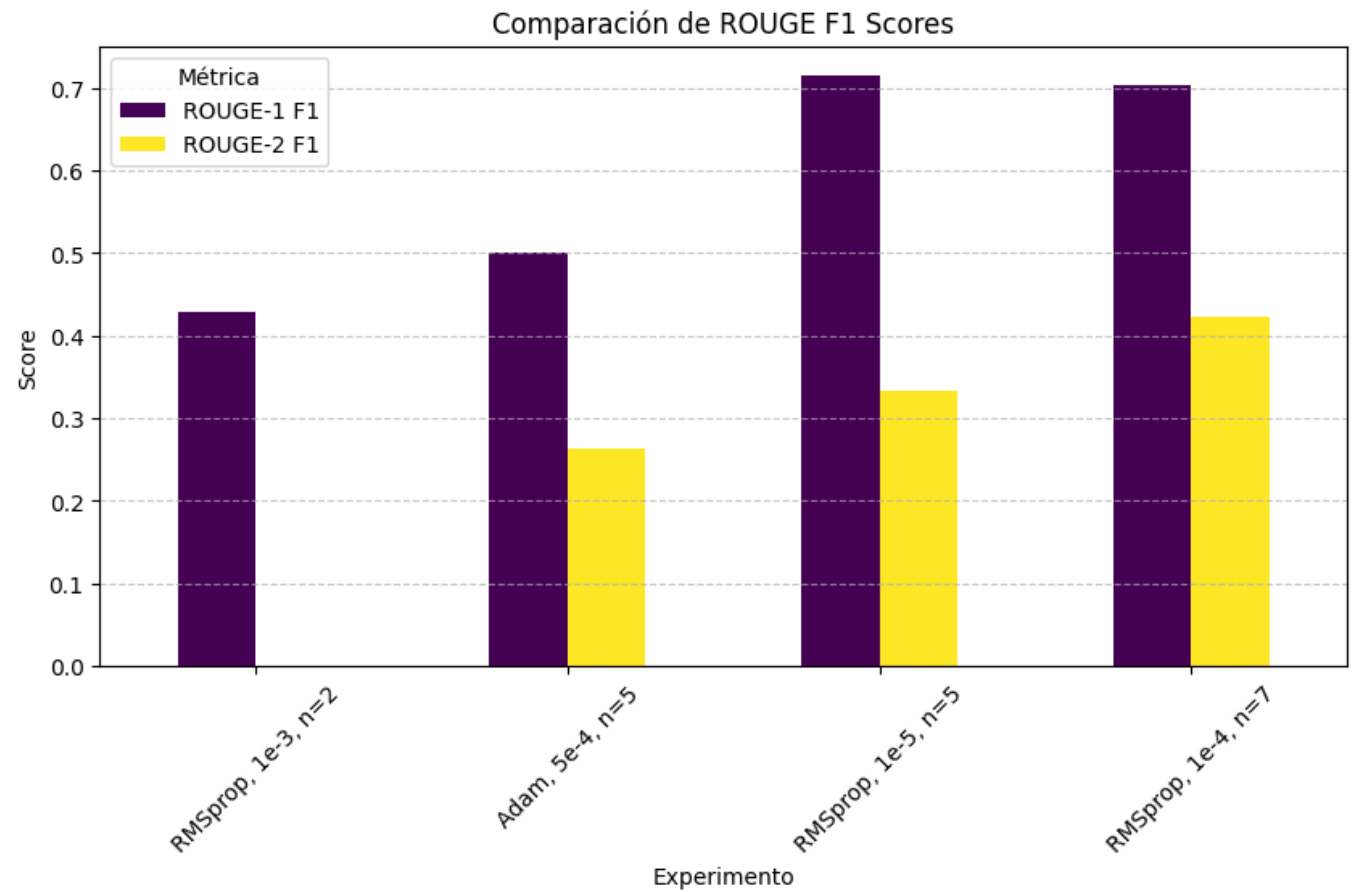
```
# Crear un DataFrame resumen con valores promedio
summary_data = {
    "Experimento": ["RMSprop, 1e-3, n=2", "Adam, 5e-4, n=5", "RMSprop, 1e-5, n=5", "RMSprop, 1e-4, n=7"],
    "ROUGE-1 F1": [
        #df_rouge_1["ROUGE-1 F1-score"].mean(),
        df_rouge_2["ROUGE-1 F1-score"].mean(),
        df_rouge_3["ROUGE-1 F1-score"].mean(),
        df_rouge_4["ROUGE-1 F1-score"].mean(),
        df_rouge_7["ROUGE-1 F1-score"].mean(),
    ],
    "ROUGE-2 F1": [
        #df_rouge_1["ROUGE-2 F1-score"].mean(),
        df_rouge_2["ROUGE-2 F1-score"].mean(),
        df_rouge_3["ROUGE-2 F1-score"].mean(),
        df_rouge_4["ROUGE-2 F1-score"].mean(),
        df_rouge_7["ROUGE-2 F1-score"].mean(),
    ],
}
```

```
}

df_summary = pd.DataFrame(summary_data)
print(df_summary)
```

	Experimento	ROUGE-1 F1	ROUGE-2 F1
0	RMSprop, 1e-3, n=2	0.428571	0.000000
1	Adam, 5e-4, n=5	0.500000	0.263158
2	RMSprop, 1e-5, n=5	0.714286	0.333333
3	RMSprop, 1e-4, n=7	0.703606	0.421978

```
fig, ax = plt.subplots(figsize=(10, 5))
df_summary.set_index("Experimento").plot(kind="bar", ax=ax, cmap="viridis")
plt.title("Comparación de ROUGE F1 Scores")
plt.ylabel("Score")
plt.xticks(rotation=45)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.legend(title="Métrica")
plt.show()
```



```
pd.set_option("display.max_colwidth",60)
pd.set_option("display.max_columns", None) # Mostrar todas las columnas
pd.set_option("display.width", 130) # Ajustar el ancho de la tabla
print(df_rouge_7.head(10)) # Imprimir las primeras 10 filas
```

Input			
Reference \			
0	Our teacher is a real idiot.	[start] Nuestra	
	profesora es una verdadera idiota. [end]		
1	Tom isn't budging on this one.	[start] Tom	
	no está cediendo sobre esto. [end]		
2	She was absorbed in her work.	[start] A	
	ella la absorbió su trabajo. [end]		
3	War broke out in 1939.		
	[start] La guerra estalló en 1939. [end]		
4	I think I understood.		
	[start] Yo creo que comprendí. [end]		
5	I need to find someone to back up Tom's alibi.	[start] Necesito	
	encontrar a alguien que confirme la coa...		
6	I don't want to kill you.		
	[start] No quiero matarte. [end]		
7	Tom causes me a lot of trouble.	[start]	
	Tom me causa muchos problemas. [end]		
8	This restaurant is full of young couples.	[start] Este restaurante	
	está lleno de parejas jóvenes. ...		
9	I'm not the only one who can do that.	[start] No soy	
	el único que puede hacer eso. [end]		
Translated ROUGE-1			
Precision	ROUGE-1 Recall	ROUGE-1 F1-score	\
0	[start] nuestro profesor es un verdadero idiota [end]		
0.500000	0.500000	0.500000	
1	[start] tom no está [UNK] esto [end]		
0.857143	0.750000	0.800000	
2	[start] ella estaba [UNK] en su trabajo [end]		
0.625000	0.625000	0.625000	
3	[start] la guerra se rompió en [UNK] [end]		
0.625000	0.714286	0.666667	
4	[start] creo que me [UNK] [end]		
0.666667	0.666667	0.666667	
5	[start] necesito encontrar a alguien a la espalda de tom...		
0.818182	0.750000	0.782609	
6	[start] no quiero [UNK] [end]		
0.800000	0.800000	0.800000	
7	[start] tom me [UNK] mucho problemas [end]		
0.857143	0.857143	0.857143	
8	[start] este restaurante está llena de jóvenes [end]		
0.888889	0.800000	0.842105	
9	[start] no soy el único que puede hacer eso [end]		
1.000000	1.000000	1.000000	

	ROUGE-2 Precision	ROUGE-2 Recall	ROUGE-2 F1-score
0	0.142857	0.142857	0.142857
1	0.666667	0.571429	0.615385
2	0.285714	0.285714	0.285714
3	0.285714	0.333333	0.307692
4	0.200000	0.200000	0.200000
5	0.600000	0.545455	0.571429
6	0.500000	0.500000	0.500000
7	0.666667	0.666667	0.666667
8	0.625000	0.555556	0.588235
9	1.000000	1.000000	1.000000

**BLEU: (Bilingual Evaluation Understudy)**

BLEU es una métrica de evaluación de la calidad de las traducciones automáticas, comparando la traducción generada por una máquina con una o varias traducciones de referencia humanas.

BLEU-1 (1-grama): Mide la precisión de los 1-gramas entre la traducción generada y la referencia. Calcula cuántas palabras individuales de la traducción generada coinciden con las palabras en la traducción de referencia.

BLEU-1 es muy sensible a la coincidencia exacta de palabras individuales, sin tener en cuenta el orden o contexto.

BLEU-2 (2-gramas): Mide la precisión de los 2-gramas entre la traducción generada y la referencia.

Calcula cuántos pares consecutivos de palabras en la traducción generada coinciden con los pares consecutivos de palabras en la traducción de referencia.

```
#df_bleu_1 = pd.read_csv("bleu_scores_1.csv") # adam, , 1e-3, ngram = 2
df_bleu_2 = pd.read_csv("bleu_scores_2.csv") # RMSprop , 1e-3, ngram = 2
df_bleu_3 = pd.read_csv("bleu_scores_3.csv") # adam, 5e-4, ngram = 5
df_bleu_4 = pd.read_csv("bleu_scores_4.csv") # RMSprop, 1e-4, ngram = 5
df_bleu_7 = pd.read_csv("bleu_scores_7.csv") # RMSprop, 1e-4, ngram = 7

# Resumen de las estadísticas descriptivas
print("Estadísticas descriptivas del DataFrame:")
print(df_bleu_7.describe())
```

Estadísticas descriptivas del DataFrame:				
	BLEU-1	BLEU-2	BLEU-3	BLEU-4
count	100.000000	100.000000	100.000000	100.000000
mean	0.501985	0.204325	0.116281	0.082318
std	0.121585	0.146589	0.111858	0.088452
min	0.268128	0.061616	0.035988	0.026510
25%	0.413343	0.084515	0.053842	0.043472
50%	0.500000	0.129099	0.081407	0.057893

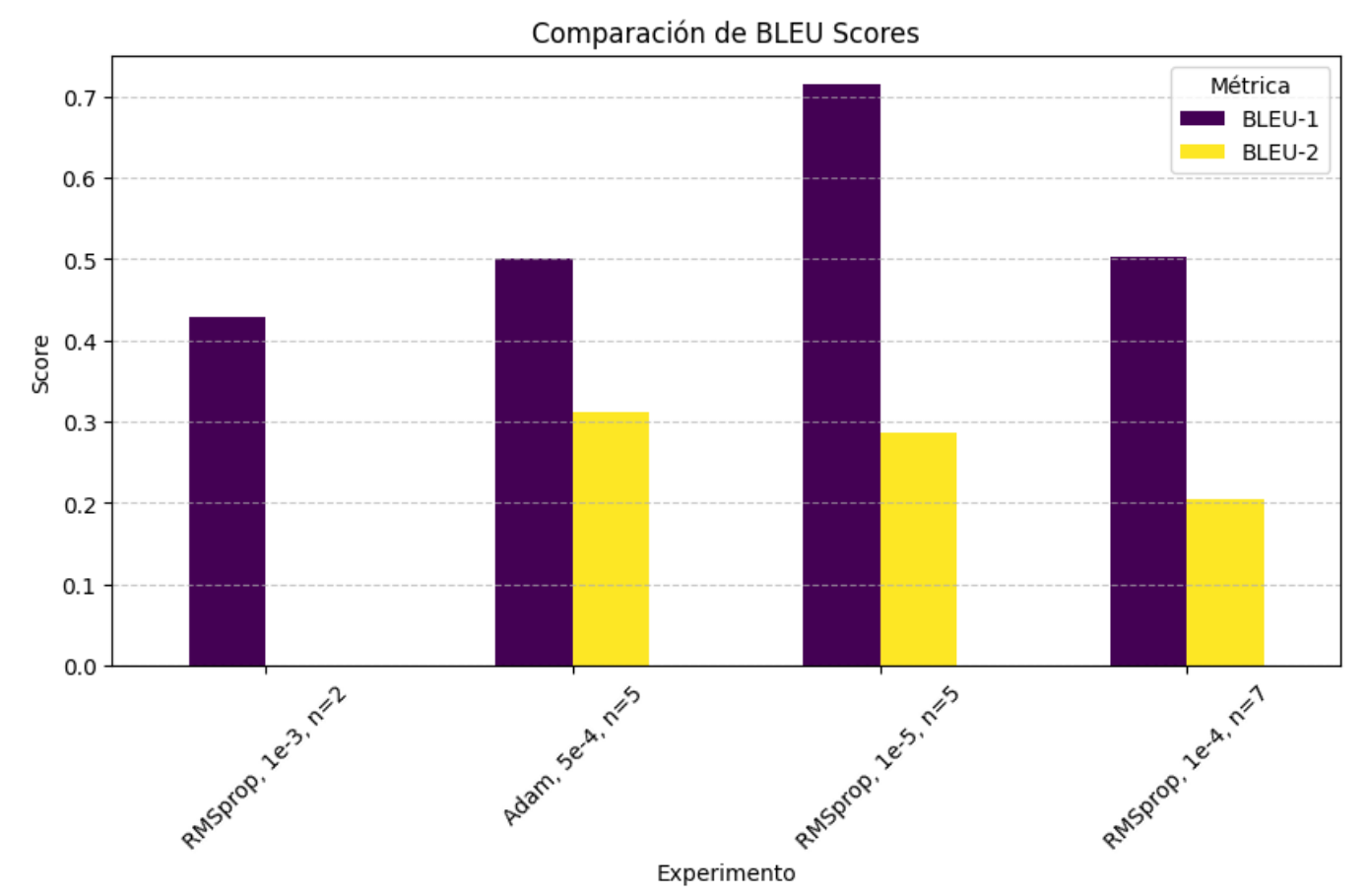
75%	0.600000	0.308607	0.126575	0.095544
max	0.800000	0.666667	0.608751	0.555524

```
# Crear un DataFrame resumen con valores promedio
summary_data = {
    "Experimento": ["RMSprop, 1e-3, n=2",
                    "Adam, 5e-4, n=5", "RMSprop, 1e-5, n=5", "RMSprop, 1e-4, n=7"],
    "BLEU-1": [
        #df_bleu_1["BLEU-1"].mean(),
        df_bleu_2["BLEU-1"].mean(),
        df_bleu_3["BLEU-1"].mean(),
        df_bleu_4["BLEU-1"].mean(),
        df_bleu_7["BLEU-1"].mean(),
    ],
    "BLEU-2": [
        #df_bleu_1["BLEU-2"].mean(),
        df_bleu_2["BLEU-2"].mean(),
        df_bleu_3["BLEU-2"].mean(),
        df_bleu_4["BLEU-2"].mean(),
        df_bleu_7["BLEU-2"].mean(),
    ],
}

df_summary = pd.DataFrame(summary_data)
print(df_summary)
```

	Experimento	BLEU-1	BLEU-2
0	RMSprop, 1e-3, n=2	0.428571	0.000000
1	Adam, 5e-4, n=5	0.500000	0.312500
2	RMSprop, 1e-5, n=5	0.714286	0.285714
3	RMSprop, 1e-4, n=7	0.501985	0.204325

```
fig, ax = plt.subplots(figsize=(10, 5))
df_summary.set_index("Experimento").plot(kind="bar", ax=ax, cmap="viridis")
plt.title("Comparación de BLEU Scores")
plt.ylabel("Score")
plt.xticks(rotation=45)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.legend(title="Métrica")
plt.show()
```



```
pd.set_option("display.max_colwidth",60)
pd.set_option("display.max_columns", None) # Mostrar todas las columnas
pd.set_option("display.width", 130) # Ajustar el ancho de la tabla
print(df_bleu_7.head(10)) # Imprimir las primeras 10 filas
```

Input		
Reference \		
0	Our teacher is a real idiot.	[start] Nuestra
	profesora es una verdadera idiota. [end]	
1	Tom isn't budging on this one.	[start] Tom
	no está cediendo sobre esto. [end]	
2	She was absorbed in her work.	[start] A
	ella la absorbió su trabajo. [end]	
3	War broke out in 1939.	
	[start] La guerra estalló en 1939. [end]	
4	I think I understood.	
	[start] Yo creo que comprendí. [end]	
5	I need to find someone to back up Tom's alibi.	[start] Necesito
	encontrar a alguien que confirme la coa...	
6	I don't want to kill you.	
	[start] No quiero matarte. [end]	
7	Tom causes me a lot of trouble.	[start]
	Tom me causa muchos problemas. [end]	
8	This restaurant is full of young couples.	[start] Este restaurante
	está lleno de parejas jóvenes. ...	

9	I'm not the only one who can do that.			[start] No soy
	el único que puede hacer eso. [end]			
			Translated	BLEU-1
BLEU-2	BLEU-3	BLEU-4		
0	[start] nuestro profesor es un verdadero idiota [end]			0.375000
0.073193	0.046106	0.036556		
1	[start] tom no está [UNK] esto [end]			0.495359
0.267524	0.109725	0.072012		
2	[start] ella estaba [UNK] en su trabajo [end]			0.500000
0.084515	0.050698	0.039281		
3	[start] la guerra se rompió en [UNK] [end]			0.500000
0.084515	0.050698	0.039281		
4	[start] creo que me [UNK] [end]			0.666667
0.365148	0.152247	0.102669		
5	[start] necesito encontrar a alguien a la espalda de tom...			0.581064
0.325751	0.223966	0.105286		
6	[start] no quiero [UNK] [end]			0.600000
0.122474	0.081407	0.070711		
7	[start] tom me [UNK] mucho problemas [end]			0.428571
0.084515	0.053842	0.043472		
8	[start] este restaurante está llena de jóvenes [end]			0.551561
0.263696	0.102963	0.065182		
9	[start] no soy el único que puede hacer eso [end]			0.800000
0.666667	0.608751	0.555524		

## Conclusiones

Al realizar modificaciones en los parámetros ngrams, optimizador, metrica o tasa de aprendizaje se apreciaron cambios significativos en las traducciones asi como en la velocidad de entrenamiento. Se compararon n-gramas de 2, 5 y 7 palabras, en combinación con dos optimizadores Adam y RMSprop. Se analizaron dos métricas, la ROUGE y BLEU. Se observo que BLEU scores no muestra un 100% de incluso cuando la traducción generada es perfecta en términos de significado, esto es ya que BLEU no mide solo la calidad semántica, sino que también depende de la coincidencia exacta de los n-gramas y otros factores que pueden hacer que no se alcance un puntaje del 100%. En cuanto a ROUGE se pueden apreciar un promedio de buenas traducciones para ROUGE-1 (1-grama) en la traducción, sin embargo baja bastante para ROUGE-2 (2-grama) esto es la comparación de pares de palabras, se observa que este falla al interpretar el género de las palabras.

Por ejemplo: oración en inglés: Our teacher is a real idiot. Referencia: Nuestra profesora es una verdadera idiota. Traducción: nuestro profesor es un verdadero idiota ROUGE-1 = 50, ROUGE-2 = 0.142857 a pesar de ser una traducción apropiada, asume el genero y aparentemente eso disminuye el score.

Por otro lado BLEU-1 = 37.5%, BLEU-2 = 7.4%

Los puntajes que arrojan estas métricas no son equivalentes y no deben interpretarse como comparables entre sí. BLEU se enfoca en la precisión de los n-gramas generados, mientras que ROUGE se enfoca en qué tanto de la referencia se recupera en la traducción generada.



BLEU tiende a ser más estricto y solo premia las coincidencias exactas, lo que puede llevar a puntajes más bajos si el modelo es flexible en su expresión.

ROUGE, siendo más flexible, generalmente genera puntuaciones más altas ya que evalúa el recall y no penaliza mucho la falta de coincidencias exactas o el cambio de palabras.

Al promediar los scores se pudo observar que para el optimizador RMS con 5 y 7 ngrams los ROUGE scores son muy similares. A medida que se incremento el número de ngrams estos scores aumentaron cuando se analizan scores para 1-grams, y 2-grams en el caso de ROUGE, mientras que los BLEU-1 scores aumentan los BLEU-2 scores disminuyen, esto parece indicar que conforme se agregan más ngrams a los textos vectorizados este score se vuelve más selectivo.

Por cuestiones de recursos no ha sido posible examinar más configuraciones de los parámetros solicitados, pero es un excelente ejercicio probar más configuraciones para determinar mejores resultados comparar con los resultados obtenidos en el grupo del diplomado. ``