Introducción

La tecnología ha avanzado muy rápido desde la llegada de la inteligencia artificial, gracias a diferentes factores es posible hoy en día tener herramientas que hace unos años eran inalcanzables. Hay muchos modelos computacionales que permiten la interacción con el usuario para desarrollar tareas complejas que va desde el análisis de datos hasta indicaciones precisas para poder resolver dichos problemas, uno de estos modelos son las Redes Neuronales Recurrentes (RNN)\cite{ibm_2023}. Las RNN son una clase de redes neuronales que permiten el procesamiento de datos de una manera más fácil ya que son capaces de retener información previa a través de sus conexiones internas haciéndolas ideales para problemas como procesamiento del lenguaje natural, ayudando a la creación de chatbots, resumir textos, traducción, entre otros.

Mediante el uso de una técnica llamada ''embedding´´ es posible representar caracteres o palabras como vectores de números reales en un espacio de alta dimensión, de esta forma es más fácil poder recordar la secuencia de los números para poder reconocer relación entre las palabras. En este reporte se quiere poder trabajar con un modelo para poder evaluar su desempeño y poder modificar diferentes parámetros y lograr observar de qué forma puede desempeñar una mejor tarea. El código se desarrollo en el lenguaje Python a través de Google Collab Pro para poder acceder a procesos computacionales más rápidos.

Análisis

Las RNN son muy usadas en el procesamiento de lenguaje natural debido a su gran capacidad de poder recordar mediante conexiones recurrentes, lo que significa que la salida de una unidad en un momento dado se retroalimenta como entrada en el siguiente paso de tiempo. La RNN actualiza en cada capa su estado interno debido a la información de las capas pasadas y dando paso a nueva información para capas futuras. Se han desarrollado diferentes variantes para poder trabajar diferentes problemas dependiendo de su naturaleza, algunos ejemplos son LSTM y GRU, estas variantes usan un gradiente de desvanecimiento diferente.

En este modelo se hace uso de Long Short-Term Memory (LSTM), las LSTM fueron creadas por Sepp Hochreiter y Jürgen Schmidhuber en 1997. El funcionamiento de las LSTM, en cada paso de tiempo, toman tres entradas: la entrada actual, la salida del paso de tiempo anterior y el estado anterior de la celda de memoria, y mediante funciones de activación, generalmente es la sigmoide. A diferencia de las RNN, estas son capaces de recordar información para intervalos de tiempo largos, como en la traducción automática\cite{LSTM1}. Las LSTM son usadas en varias cosas donde generalmente es crucial recodar grandes cantidades de caracteres.

Las LSTM hoy en día son muy usadas debido a su manejo efectivo de dependencias a largo plazo, su amplia versatilidad en diferentes aplicaciones, su fácil adaptabilidad con nuevas tecnologías emergentes, que es un modelo avanzado, y que constantemente hay investigación alrededor de las LSTM para poder mejorarlas.

Se espera poder desarrollar un modelo que funcione y se pueden modificar sus parámetros para poder obtener un modelo mejor optimizado, dentro de las aplicacionesm son muy grandes, un claro ejemplo es ChatGPT, donde puedes tener una plática y puedes solicitar diferentes tareas; otro ejemplo son los chatbots, estos tienen menos capacidad para realizar tareas, pero funcionan como una interfaz entre el usuario y toda la computación por detrás para poder resolver algún problema.

In [ ]:
```
import tensorflow
from tensorflow import keras
import torchtext
import gc
```

In [ ]:
```
!pip install portalocker
```

```
Requirement already satisfied: portalocker in /usr/local/lib/python3.10/dist-packages (2.8.2)
```

Diseño

La preparación de datos es fundamental para poder meter los datos a un modelo, sirven principalmente para limpiar los datos y ordenarlos en un formato adecuado donde sea más fácil el procesamiento, por ejemplo la división en varios conjuntos, de entrenamiento y validación.

Primeor se cargan los datos desde un conjunto de datos WikiText-2. Este conjunto de datos incluye conjuntos de entrenamiento, validación y prueba.

In [ ]:
```
train_dataset, valid_dataset, test_dataset = torchtext.datasets.WikiText2()
```

Se crea una lista donde están todos los textos del conjunto de entrenamiento y se calcula la longitud de textos del conjunto de entrenamiento, son 36178

In [ ]:
```
X_train_text = [text for text in train_dataset]
len(X_train_text)
```

Out[ ]:
```
36718
```

Se crea un tokenizador para poder introducirlo en el set de entrenamientos, para poder construir un vocabulario de caracteres

In [ ]:
```
from keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer(char_level=True)

tokenizer.fit_on_texts(X_train_text)
```

In [ ]:
```
print(tokenizer.word_index)
```

```
{' ': 1, 'e': 2, 't': 3, 'a': 4, 'n': 5, 'i': 6, 'o': 7, 'r': 8, 's': 9, 'h':
10, 'd': 11, 'l': 12, 'u': 13, 'c': 14, 'm': 15, 'f': 16, 'g': 17, 'p': 18,
'w': 19, 'b': 20, 'y': 21, 'k': 22, ',': 23, '.': 24, 'v': 25, '<': 26, '>':
27, '@': 28, '\n': 29, '1': 30, '0': 31, '=': 32, '"': 33, '2': 34, "'": 35,
'9': 36, '-': 37, 'j': 38, 'x': 39, ')': 40, '(': 41, '3': 42, '5': 43, '8':
44, '4': 45, '6': 46, '7': 47, 'z': 48, 'q': 49, ';': 50, '—': 51, ':': 52,
'/': 53, '–': 54, '%': 55, 'é': 56, '$': 57, '[': 58, ']': 59, '&': 60, '!':
61, 'í': 62, '’': 63, 'á': 64, 'ā': 65, '£': 66, '°': 67, '?': 68, 'ó': 69,
'+': 70, '#': 71, 'š': 72, '−': 73, 'ō': 74, 'ö': 75, 'è': 76, '×': 77, 'ü':
78, 'ä': 79, '‘': 80, 'ś': 81, 'ć': 82, '“': 83, 'ø': 84, 'ł': 85, 'ç': 86,
'”': 87, '₹': 88, 'ã': 89, 'µ': 90, 'ì': 91, 'ư': 92, '\ufeff': 93, 'æ': 94,
'…': 95, '→': 96, 'ơ': 97, 'ñ': 98, 'å': 99, '☉': 100, '‘': 101, '~': 102,
'∗': 103, '⁄': 104, 'î': 105, '²': 106, 'ë': 107, 'ệ': 108, 'ī': 109, 'ú': 11
0, 'ễ': 111, 'ô': 112, 'à': 113, 'ū': 114, 'ă': 115, '^': 116, '♯': 117, 'ê':
118, '‐': 119, 'ỳ': 120, 'đ': 121, 'μ': 122, '≤': 123, '125 :'～' ,124 :'ل',
'ṃ': 126, '†': 127, '€': 128, 'ួ': 129, '・': 130, '±': 131, 'ž': 132, 'ė': 1
33, '〈': 134, '〉': 135, 'β': 136, 'č': 137, 'α': 138, 'û': 139, '♭': 140,
'½': 141, '„': 142, 'ị': 143, 'c': 144, 'ṭ': 145, 'γ': 146, 'â': 147, '′': 14
8, '大': 149, '空': 150, '̃': 151, 'ớ': 152, 'ầ': 153, '⅓': 154, ',155 :'ا
156 :'ہ', 'ṅ': 157, '჻': 158, 'ṯ': 159, 'ş': 160, '16 :'ن' ,162 :'ص' ,161 :'ح
167 :'¡' ,166 :'¥' ,165 :'³' ,164 :'″' ,3, 'ấ': 168, 'ả': 169, '火': 170,
'礮': 171, '·': 172, 'ḥ': 173, 'ჯ': 174, 'ი': 175, 'ო': 176, 'ხ': 177, 'ვ': 1
78, 'კ': 179, '戦': 180, '場': 181, 'の': 182, 'ヴ': 183, 'ァ': 184, 'ル': 185,
'キ': 186, 'ュ': 187, 'リ': 188, 'ア': 189, '₤': 190, 'ż': 191, 'ń': 192, '̇':
193, 'ง': 194, 'ก': 195, '์': 196, 'ล': 197, 'ย': 198, 'า': 199, 'ณ': 200,
'ม': 201, 'ั': 202, 'ต': 203, 'ร': 204, '่': 205, '§': 206, 'ス': 207, 'ト': 20
8, 'ッ': 209, 'プ': 210, 'ʿ': 211, 'þ': 212, '\\': 213, '`': 214, '⅔': 215,
'ắ': 216, 'ử': 217, '|': 218, '攻': 219, '殻': 220, '機': 221, '動': 222,
'隊': 223, 'ų': 224, 'к': 225, 'ò': 226, 'о': 227, 'в': 228, 'е': 229, 'т': 2
30, 'κ': 231, 'а': 232, 'я': 233, 'ṣ': 234, 'დ': 235, 'უ': 236, 'ბ': 237,
'რ': 238, 'ს': 239, 'ძ': 240, 'წ': 241, 'გ': 242, 'ც': 243}
```

En este paso, se itera sobre los diferentes textos y crea secuencias de entrada de longitud y su etiqueta correspondiente y se añaden a los sets de entrenamiento para finalmente convertirlas en arrys de Numpy para poder usarlo más fácilmente en el entrenamiento del modelo. Se usa una longitud de secuencia de 100 caracteres inicialmente

In [ ]:
```
%%time

import numpy as np
train_dataset, valid_dataset, test_dataset = torchtext.datasets.WikiText2()
```

```
seq_length = 100
X_train, Y_train = [], []

for text in X_train_text[:6000]:
    for i in range(0, len(text)-seq_length):
        inp_seq = text[i:i+seq_length].lower()
        out_seq = text[i+seq_length].lower()
        X_train.append(inp_seq)
        Y_train.append(tokenizer.word_index[out_seq])

X_train = tokenizer.texts_to_sequences(X_train)

X_train, Y_train = np.array(X_train, dtype=np.int32), np.array(Y_train)

X_train.shape, Y_train.shape
```

```
CPU times: user 28.9 s, sys: 630 ms, total: 29.5 s
Wall time: 29.4 s
```

Out[ ]:

```
((1377719, 100), (1377719,))
```

Se puede observar que primero se declara la longitud de embedding, es decir, la longitud de los vectores para cada palabra; después viene la dimensión de unidades de la capa LSTM. Por parte del modelo, consta por 4 capas, la primera es una capa de Embedding la cual convierte los índices de palabras en vectores de tamaño, donde el tamaño del vocabulario está declarado por "input_dim" y se le suma 1 y la extensión del vector de entrada está declarado por "input_length"; después continuan dos capas de LSTM, la primera que sirve para poder devolver la secuencia completa a la siguiente capa en cada instante, lo cuál es muy relevante para la conexción entre las capas, y la segunda capa, a diferencia de la anterior, devuelve únicamente la salida del último paso; por último viene una capa Dense, la cual funciona como una capa de salida con la función de activación softmax que sirve para poder obetener una distribución de probabilidad sobre las palabras de vocabulario.

In [ ]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Embedding

embed_len = 50
lstm_out = 256

model = Sequential([
                    Embedding(input_dim=len(tokenizer.word_index)+1, output_
                              input_length=seq_length),
                    LSTM(lstm_out, return_sequences=True),
                    LSTM(lstm_out),
                    Dense(len(tokenizer.word_index)+1, activation="softmax")
                ])


model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 100, 50)           12200

 lstm_6 (LSTM)               (None, 100, 256)          314368

 lstm_7 (LSTM)               (None, 256)               525312

 dense_3 (Dense)             (None, 244)               62708

=================================================================
Total params: 914588 (3.49 MB)
Trainable params: 914588 (3.49 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Se hacen callbacks para poder seguir el rendimiento del modelo en cada epoch y así se puede guardar el modelo usado. Se almacena en un archivo llamado "mi_modelo_checkpoint.hdf5". Y el modelo se guardará solo si en cada monmento va mejorando

In [ ]:
```
from tensorflow.keras.callbacks import ModelCheckpoint

filepath = "mi_modelo_checkpoint.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1, save_best_
```

Se compila le modelo con un optimizaro de Adam y se declara una tasa de aprendizaje de 0.001 junto con el parámetro de pérdida "sparse_categorical_crossentropy".

In [ ]:
```
from tensorflow.keras.optimizers import Adam
from keras import backend as K

model.compile(optimizer=Adam(learning_rate=0.001), loss="sparse_categorical_
```

Se empieza a hacer las iteraciones de entrenamiento separando el set de entrenamiento en un 20% para validación, se decidió usar un 20% porque 20 caracteres parecen una longitud correcta refiriéndose a una oración corta. Se usa un batch_size de 1025 y 50 iteraciones con el callback que se definió anteriormente

In [ ]:
```
history = model.fit(X_train, Y_train, validation_split=0.2, batch_size=1024,
```

```
Epoch 1/50
1076/1077 [============================>.] - ETA: 0s - loss: 2.5480
Epoch 1: loss improved from inf to 2.54777, saving model to mi_modelo_checkpo
int.hdf5
1077/1077 [==============================] - 81s 73ms/step - loss: 2.5478 - v
al_loss: 2.1209
Epoch 2/50
   2/1077 [..............................] - ETA: 1:10 - loss: 2.1165
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: Us
erWarning: You are saving your model as an HDF5 file via `model.save()`. This
file format is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

```
1077/1077 [==============================] - ETA: 0s - loss: 2.0373
Epoch 2: loss improved from 2.54777 to 2.03734, saving model to mi_modelo_che
ckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 2.0373 - v
al_loss: 1.9493
Epoch 3/50
1077/1077 [==============================] - ETA: 0s - loss: 1.8554
Epoch 3: loss improved from 2.03734 to 1.85543, saving model to mi_modelo_che
ckpoint.hdf5
1077/1077 [==============================] - 77s 72ms/step - loss: 1.8554 - v
al_loss: 1.8489
Epoch 4/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.7594
Epoch 4: loss improved from 1.85543 to 1.75940, saving model to mi_modelo_che
ckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.7594 - v
al_loss: 1.7959
Epoch 5/50
1077/1077 [==============================] - ETA: 0s - loss: 1.6964
Epoch 5: loss improved from 1.75940 to 1.69643, saving model to mi_modelo_che
ckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.6964 - v
al_loss: 1.7366
Epoch 6/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.6313
Epoch 6: loss improved from 1.69643 to 1.63135, saving model to mi_modelo_che
ckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.6313 - v
al_loss: 1.7005
Epoch 7/50
1077/1077 [==============================] - ETA: 0s - loss: 1.5803
Epoch 7: loss improved from 1.63135 to 1.58034, saving model to mi_modelo_che
ckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.5803 - v
al_loss: 1.6573
Epoch 8/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5403
Epoch 8: loss improved from 1.58034 to 1.54032, saving model to mi_modelo_che
ckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.5403 - v
al_loss: 1.6308
Epoch 9/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5072
Epoch 9: loss improved from 1.54032 to 1.50717, saving model to mi_modelo_che
ckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.5072 - v
al_loss: 1.6076
Epoch 10/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4776
Epoch 10: loss improved from 1.50717 to 1.47757, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.4776 - v
al_loss: 1.5840
Epoch 11/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4519
Epoch 11: loss improved from 1.47757 to 1.45190, saving model to mi_modelo_ch
```

```
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.4519 - v
al_loss: 1.5782
Epoch 12/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4309
Epoch 12: loss improved from 1.45190 to 1.43091, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.4309 - v
al_loss: 1.5660
Epoch 13/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4090
Epoch 13: loss improved from 1.43091 to 1.40903, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.4090 - v
al_loss: 1.5494
Epoch 14/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3901
Epoch 14: loss improved from 1.40903 to 1.38998, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.3900 - v
al_loss: 1.5425
Epoch 15/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3809
Epoch 15: loss improved from 1.38998 to 1.38092, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.3809 - v
al_loss: 1.5342
Epoch 16/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3659
Epoch 16: loss improved from 1.38092 to 1.36586, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.3659 - v
al_loss: 1.5197
Epoch 17/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3462
Epoch 17: loss improved from 1.36586 to 1.34618, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.3462 - v
al_loss: 1.5103
Epoch 18/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3339
Epoch 18: loss improved from 1.34618 to 1.33385, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.3339 - v
al_loss: 1.5149
Epoch 19/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3481
Epoch 19: loss did not improve from 1.33385
1077/1077 [==============================] - 77s 71ms/step - loss: 1.3481 - v
al_loss: 1.5026
Epoch 20/50
1077/1077 [==============================] - ETA: 0s - loss: 1.3387
Epoch 20: loss did not improve from 1.33385
1077/1077 [==============================] - 77s 71ms/step - loss: 1.3387 - v
al_loss: 1.4975
Epoch 21/50
```

```
1076/1077 [============================>.] - ETA: 0s - loss: 1.3227
Epoch 21: loss improved from 1.33385 to 1.32269, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.3227 - v
al_loss: 1.4840
Epoch 22/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3089
Epoch 22: loss improved from 1.32269 to 1.30891, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.3089 - v
al_loss: 1.4757
Epoch 23/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2973
Epoch 23: loss improved from 1.30891 to 1.29723, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.2972 - v
al_loss: 1.4767
Epoch 24/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2861
Epoch 24: loss improved from 1.29723 to 1.28613, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.2861 - v
al_loss: 1.4646
Epoch 25/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2761
Epoch 25: loss improved from 1.28613 to 1.27611, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.2761 - v
al_loss: 1.4628
Epoch 26/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2663
Epoch 26: loss improved from 1.27611 to 1.26629, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.2663 - v
al_loss: 1.4581
Epoch 27/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2575
Epoch 27: loss improved from 1.26629 to 1.25751, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.2575 - v
al_loss: 1.4522
Epoch 28/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2489
Epoch 28: loss improved from 1.25751 to 1.24889, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.2489 - v
al_loss: 1.4514
Epoch 29/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2411
Epoch 29: loss improved from 1.24889 to 1.24111, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.2411 - v
al_loss: 1.4484
Epoch 30/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2336
Epoch 30: loss improved from 1.24111 to 1.23354, saving model to mi_modelo_ch
```

```
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.2335 - v
al_loss: 1.4491
Epoch 31/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2265
Epoch 31: loss improved from 1.23354 to 1.22653, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.2265 - v
al_loss: 1.4440
Epoch 32/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2196
Epoch 32: loss improved from 1.22653 to 1.21955, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.2196 - v
al_loss: 1.4390
Epoch 33/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2126
Epoch 33: loss improved from 1.21955 to 1.21267, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.2127 - v
al_loss: 1.4424
Epoch 34/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2065
Epoch 34: loss improved from 1.21267 to 1.20653, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.2065 - v
al_loss: 1.4400
Epoch 35/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2002
Epoch 35: loss improved from 1.20653 to 1.20023, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.2002 - v
al_loss: 1.4357
Epoch 36/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1943
Epoch 36: loss improved from 1.20023 to 1.19426, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1943 - v
al_loss: 1.4372
Epoch 37/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1888
Epoch 37: loss improved from 1.19426 to 1.18880, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1888 - v
al_loss: 1.4394
Epoch 38/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1832
Epoch 38: loss improved from 1.18880 to 1.18318, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1832 - v
al_loss: 1.4322
Epoch 39/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1777
Epoch 39: loss improved from 1.18318 to 1.17768, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1777 - v
```

```
al_loss: 1.4343
Epoch 40/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1725
Epoch 40: loss improved from 1.17768 to 1.17254, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1725 - v
al_loss: 1.4357
Epoch 41/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1674
Epoch 41: loss improved from 1.17254 to 1.16737, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1674 - v
al_loss: 1.4358
Epoch 42/50
1077/1077 [==============================] - ETA: 0s - loss: 1.1622
Epoch 42: loss improved from 1.16737 to 1.16218, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1622 - v
al_loss: 1.4340
Epoch 43/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1576
Epoch 43: loss improved from 1.16218 to 1.15761, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1576 - v
al_loss: 1.4317
Epoch 44/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1524
Epoch 44: loss improved from 1.15761 to 1.15240, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1524 - v
al_loss: 1.4388
Epoch 45/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1482
Epoch 45: loss improved from 1.15240 to 1.14820, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1482 - v
al_loss: 1.4400
Epoch 46/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1431
Epoch 46: loss improved from 1.14820 to 1.14309, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1431 - v
al_loss: 1.4397
Epoch 47/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1391
Epoch 47: loss improved from 1.14309 to 1.13913, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1391 - v
al_loss: 1.4420
Epoch 48/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1344
Epoch 48: loss improved from 1.13913 to 1.13443, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1344 - v
al_loss: 1.4426
Epoch 49/50
```

```
1076/1077 [============================>.] - ETA: 0s - loss: 1.1303
Epoch 49: loss improved from 1.13443 to 1.13033, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1303 - v
al_loss: 1.4459
Epoch 50/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.1260
Epoch 50: loss improved from 1.13033 to 1.12602, saving model to mi_modelo_ch
eckpoint.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.1260 - v
al_loss: 1.4424
```

Se guarda el historial en un archivo tipo json.

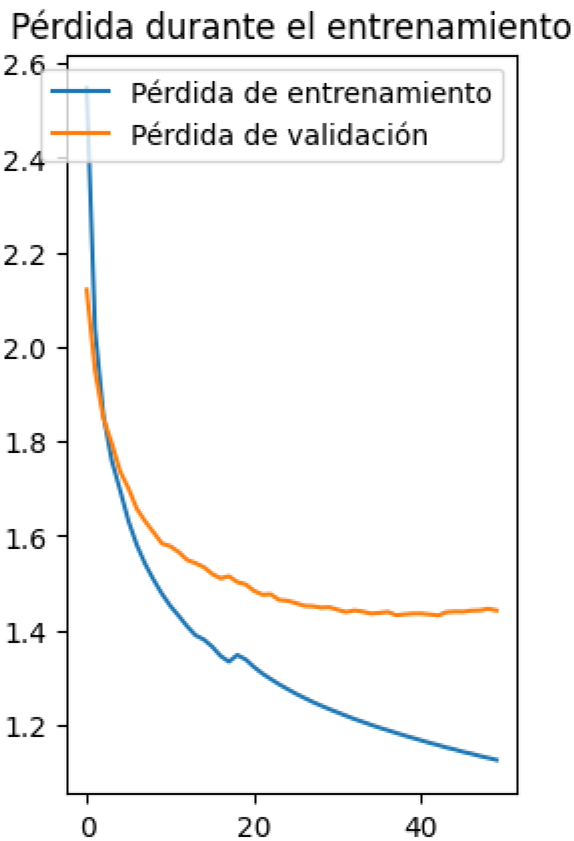
In [ ]:
```python
import json
with open('historial_entrenamiento.json', 'w') as f:
    json.dump(history.history, f)
```

Se grafica la pérdida de ambos set de entrenamiento, se puede observar como para el set de entrenamiento disminuye a más de 1.2 pero para el de validación se puede percibir cómo va aumentando nuevamente indicando overfitting

In [ ]:
```python
import matplotlib.pyplot as plt

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Pérdida de entrenamiento')
plt.plot(history.history['val_loss'], label='Pérdida de validación')
plt.title('Pérdida durante el entrenamiento')
plt.legend()

plt.show()
```



Se quiere poner a prueba el modelo, primero se declara una semilla aleatoria para poder hacer un for para generar 100 caracteres mediante la probabilidad y lo añade a la lista

In [ ]:
```
import random

random.seed(123)
idx = random.randint(0, len(X_train))
pattern = X_train[idx].flatten().tolist()

print("Initial Pattern : {}".format("".join([tokenizer.index_word[idx] for i

generated_text = []
for i in range(100):
    X_batch = np.array(pattern, dtype=np.int32).reshape(1, seq_length) ## De
    preds = model.predict(X_batch) ## Make Prediction
    predicted_index = preds.argmax(axis=-1)[0] ## Retrieve token index
    generated_text.append(predicted_index) ## Add token index to result
    pattern.append(predicted_index) ## Add token index to original pattern
    pattern = pattern[1:] ## Resize pattern to bring again to seq_length ler

print("Generated Text : {}".format("".join([tokenizer.index_word[idx] for ic
```

```
Initial Pattern : 1987 – 88 season where he was named the ihl 's co @–@ rooki
e of the year and most valuable player af
1/1 [==============================] - 1s 580ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
```

```
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
Generated Text : ter the state of the second construction of the second produ
cer , and the construction of the second
```

Se puede observar el texto predecido con un poco de coherencia pero con palabras que sí existen.

Se agrega dropout

Se quiere hacer un modelo con 2 capas de dropout. El embedding length y la dimensión del LSTM se mantiene igual. Consiste en una red neuronal similar pero agregando 2 capas de Dropout, cada una después de cada capa LSTM. Esto para ayudar a prevenir el sobreajuste disminuyendo aleatoriamente un porcentaje de las conexiones durante el entrenamiento. De igual forma se guarda el modelo.

In [ ]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Embedding, Dropout
from tensorflow.keras.optimizers import Adam
from keras import backend as K
from tensorflow.keras.callbacks import ModelCheckpoint
import json

embed_len = 50
lstm_out = 256
dropout_rate = 0.5

model1 = Sequential([
    Embedding(input_dim=len(tokenizer.word_index)+1, output_dim=embed_len,
                              input_length=seq_length),
    LSTM(lstm_out, return_sequences=True),
    Dropout(dropout_rate),
    LSTM(lstm_out),
    Dropout(dropout_rate),
    Dense(len(tokenizer.word_index)+1, activation="softmax")
])


filepath = "mi_modelo_checkpoint_dropout.hdf5"

checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1, save_best_


model1.compile(optimizer=Adam(learning_rate=0.001), loss="sparse_categorical
history1 = model1.fit(X_train, Y_train, validation_split=0.2, batch_size=102

with open('historial_entrenamiento_dropout.json', 'w') as f:
    json.dump(history1.history, f)
```

```
Epoch 1/50
1077/1077 [==============================] - ETA: 0s - loss: 2.5609
Epoch 1: loss improved from inf to 2.56095, saving model to mi_modelo_checkpo
int_dropout.hdf5
1077/1077 [==============================] - 87s 77ms/step - loss: 2.5609 - v
al_loss: 2.1224
Epoch 2/50
   2/1077 [..............................] - ETA: 1:10 - loss: 2.2064
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: Us
erWarning: You are saving your model as an HDF5 file via `model.save()`. This
file format is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

```
1076/1077 [============================>.] - ETA: 0s - loss: 2.0788
Epoch 2: loss improved from 2.56095 to 2.07879, saving model to mi_modelo_che
ckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 2.0788 - v
al_loss: 1.9414
Epoch 3/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.9686
Epoch 3: loss improved from 2.07879 to 1.96853, saving model to mi_modelo_che
ckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.9685 - v
al_loss: 1.8532
Epoch 4/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.8787
Epoch 4: loss improved from 1.96853 to 1.87873, saving model to mi_modelo_che
ckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.8787 - v
al_loss: 1.7907
Epoch 5/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.8239
Epoch 5: loss improved from 1.87873 to 1.82392, saving model to mi_modelo_che
ckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.8239 - v
al_loss: 1.7381
Epoch 6/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.7804
Epoch 6: loss improved from 1.82392 to 1.78036, saving model to mi_modelo_che
ckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.7804 - v
al_loss: 1.7010
Epoch 7/50
1077/1077 [==============================] - ETA: 0s - loss: 1.7447
Epoch 7: loss improved from 1.78036 to 1.74468, saving model to mi_modelo_che
ckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.7447 - v
al_loss: 1.6674
Epoch 8/50
1077/1077 [==============================] - ETA: 0s - loss: 1.7144
Epoch 8: loss improved from 1.74468 to 1.71435, saving model to mi_modelo_che
ckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.7144 - v
al_loss: 1.6415
Epoch 9/50
1077/1077 [==============================] - ETA: 0s - loss: 1.7171
Epoch 9: loss did not improve from 1.71435
1077/1077 [==============================] - 78s 73ms/step - loss: 1.7171 - v
al_loss: 1.6292
Epoch 10/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.6757
Epoch 10: loss improved from 1.71435 to 1.67568, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.6757 - v
al_loss: 1.6090
Epoch 11/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.6570
Epoch 11: loss improved from 1.67568 to 1.65694, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
```

```
1077/1077 [==============================] - 78s 73ms/step - loss: 1.6569 - v
al_loss: 1.5930
Epoch 12/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.6412
Epoch 12: loss improved from 1.65694 to 1.64120, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.6412 - v
al_loss: 1.5783
Epoch 13/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.6264
Epoch 13: loss improved from 1.64120 to 1.62635, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.6264 - v
al_loss: 1.5663
Epoch 14/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.6120
Epoch 14: loss improved from 1.62635 to 1.61201, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.6120 - v
al_loss: 1.5555
Epoch 15/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.6004
Epoch 15: loss improved from 1.61201 to 1.60034, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.6003 - v
al_loss: 1.5456
Epoch 16/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5889
Epoch 16: loss improved from 1.60034 to 1.58887, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5889 - v
al_loss: 1.5360
Epoch 17/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5779
Epoch 17: loss improved from 1.58887 to 1.57791, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5779 - v
al_loss: 1.5288
Epoch 18/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5683
Epoch 18: loss improved from 1.57791 to 1.56829, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5683 - v
al_loss: 1.5201
Epoch 19/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5590
Epoch 19: loss improved from 1.56829 to 1.55902, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5590 - v
al_loss: 1.5165
Epoch 20/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5517
Epoch 20: loss improved from 1.55902 to 1.55163, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5516 - v
al_loss: 1.5099
```

```
Epoch 21/50
1077/1077 [==============================] - ETA: 0s - loss: 1.5436
Epoch 21: loss improved from 1.55163 to 1.54361, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.5436 - v
al_loss: 1.5058
Epoch 22/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5377
Epoch 22: loss improved from 1.54361 to 1.53763, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5376 - v
al_loss: 1.5004
Epoch 23/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5307
Epoch 23: loss improved from 1.53763 to 1.53068, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5307 - v
al_loss: 1.4938
Epoch 24/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5238
Epoch 24: loss improved from 1.53068 to 1.52380, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5238 - v
al_loss: 1.4887
Epoch 25/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5170
Epoch 25: loss improved from 1.52380 to 1.51698, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5170 - v
al_loss: 1.4862
Epoch 26/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5120
Epoch 26: loss improved from 1.51698 to 1.51195, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5120 - v
al_loss: 1.4808
Epoch 27/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5167
Epoch 27: loss did not improve from 1.51195
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5168 - v
al_loss: 1.4827
Epoch 28/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5029
Epoch 28: loss improved from 1.51195 to 1.50292, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.5029 - v
al_loss: 1.4745
Epoch 29/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4970
Epoch 29: loss improved from 1.50292 to 1.49700, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.4970 - v
al_loss: 1.4710
Epoch 30/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4950
Epoch 30: loss improved from 1.49700 to 1.49497, saving model to mi_modelo_ch
```

```
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.4950 - v
al_loss: 1.4698
Epoch 31/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4910
Epoch 31: loss improved from 1.49497 to 1.49099, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.4910 - v
al_loss: 1.4684
Epoch 32/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4885
Epoch 32: loss improved from 1.49099 to 1.48846, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.4885 - v
al_loss: 1.4643
Epoch 33/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4838
Epoch 33: loss improved from 1.48846 to 1.48378, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4838 - v
al_loss: 1.4610
Epoch 34/50
1077/1077 [==============================] - ETA: 0s - loss: 1.4811
Epoch 34: loss improved from 1.48378 to 1.48109, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4811 - v
al_loss: 1.4614
Epoch 35/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4780
Epoch 35: loss improved from 1.48109 to 1.47796, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4780 - v
al_loss: 1.4584
Epoch 36/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4755
Epoch 36: loss improved from 1.47796 to 1.47547, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4755 - v
al_loss: 1.4577
Epoch 37/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4723
Epoch 37: loss improved from 1.47547 to 1.47231, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4723 - v
al_loss: 1.4561
Epoch 38/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4684
Epoch 38: loss improved from 1.47231 to 1.46842, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.4684 - v
al_loss: 1.4543
Epoch 39/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4649
Epoch 39: loss improved from 1.46842 to 1.46495, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.4649 - v
```

```
al_loss: 1.4533
Epoch 40/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4616
Epoch 40: loss improved from 1.46495 to 1.46162, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4616 - v
al_loss: 1.4483
Epoch 41/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4594
Epoch 41: loss improved from 1.46162 to 1.45935, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4594 - v
al_loss: 1.4473
Epoch 42/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4568
Epoch 42: loss improved from 1.45935 to 1.45678, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4568 - v
al_loss: 1.4473
Epoch 43/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4542
Epoch 43: loss improved from 1.45678 to 1.45426, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4543 - v
al_loss: 1.4462
Epoch 44/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4520
Epoch 44: loss improved from 1.45426 to 1.45194, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4519 - v
al_loss: 1.4448
Epoch 45/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4492
Epoch 45: loss improved from 1.45194 to 1.44925, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4492 - v
al_loss: 1.4428
Epoch 46/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4476
Epoch 46: loss improved from 1.44925 to 1.44759, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 78s 73ms/step - loss: 1.4476 - v
al_loss: 1.4402
Epoch 47/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4458
Epoch 47: loss improved from 1.44759 to 1.44578, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4458 - v
al_loss: 1.4414
Epoch 48/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4423
Epoch 48: loss improved from 1.44578 to 1.44232, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4423 - v
al_loss: 1.4383
Epoch 49/50
```

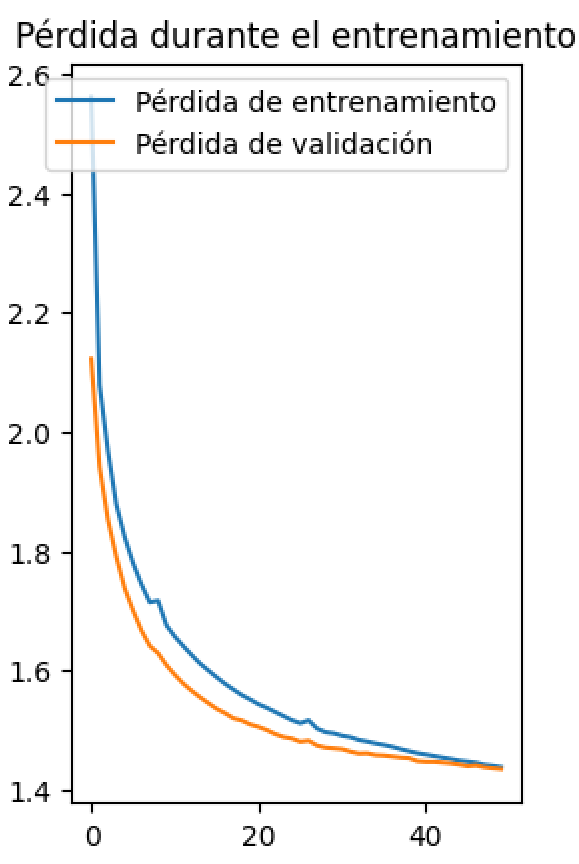
```
1076/1077 [============================>.] - ETA: 0s - loss: 1.4407
Epoch 49: loss improved from 1.44232 to 1.44067, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4407 - v
al_loss: 1.4366
Epoch 50/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4385
Epoch 50: loss improved from 1.44067 to 1.43849, saving model to mi_modelo_ch
eckpoint_dropout.hdf5
1077/1077 [==============================] - 79s 73ms/step - loss: 1.4385 - v
al_loss: 1.4352
```

Se grafica la pérdida para poder ver compararla con la anterior. En la gráfica se puede observar cómo convergen en un punto ambos sets lo que significa que no hay un sobreajuste como en el modelo anterior pero la pérdida no baja tanto como el modelo, lo que puede ser señal que no aprende tanto como el anterior.

In [ ]:
```python
import matplotlib.pyplot as plt

plt.subplot(1, 2, 2)
plt.plot(history1.history['loss'], label='Pérdida de entrenamiento')
plt.plot(history1.history['val_loss'], label='Pérdida de validación')
plt.title('Pérdida durante el entrenamiento')
plt.legend()

plt.show()
```



In [ ]:
```
import random

random.seed(123)
idx = random.randint(0, len(X_train))
pattern = X_train[idx].flatten().tolist()

print("Initial Pattern : {}".format("".join([tokenizer.index_word[idx] for i

generated_text = []
for i in range(100):
    X_batch = np.array(pattern, dtype=np.int32).reshape(1, seq_length)
    preds = model1.predict(X_batch)
    predicted_index = preds.argmax(axis=-1)[0]
    generated_text.append(predicted_index)
    pattern.append(predicted_index)
    pattern = pattern[1:]

print("Generated Text : {}".format("".join([tokenizer.index_word[idx] for ic
```

```
Initial Pattern : 1987 – 88 season where he was named the ihl 's co @-@ rooki
e of the year and most valuable player af
1/1 [==============================] - 1s 602ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
```

```
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
Generated Text : ter the season .
ather and the second state of the season of the season of the season , and th
e sea
```

Se puede observar como se forman las palabras pero igualmente sin sentido, se puede ver cómo se repiten las palabras por lo que puede ser debido a que no aprendió bien.

Con 150 palabras y learning rate de 0.0005

Se continuo con la primera red neuronal debido a que había una pérdida mayor y se quiso arreglar el tema del sobreajuste, es por eso que se decidió usar 150 palabras y usar una tasa de aprendizaje menor para que pueda aprender mejor. Primero se preparan los datos nuevamente con 150 palabras.

In [ ]:
```
%%time

import numpy as np
train_dataset, valid_dataset, test_dataset = torchtext.datasets.WikiText2()

seq_length = 150
X_train, Y_train = [], []

for text in X_train_text[:6000]:
    for i in range(0, len(text)-seq_length):
        inp_seq = text[i:i+seq_length].lower()
        out_seq = text[i+seq_length].lower()
        X_train.append(inp_seq)
        Y_train.append(tokenizer.word_index[out_seq])

X_train = tokenizer.texts_to_sequences(X_train)

X_train, Y_train = np.array(X_train, dtype=np.int32), np.array(Y_train)

X_train.shape, Y_train.shape
```

```
CPU times: user 38.6 s, sys: 847 ms, total: 39.4 s
Wall time: 39.3 s
```

Out[ ]: ((1260002, 150), (1260002,))

Como se mencionó, se usa el primer modelo.

In [ ]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Embedding, Dropout
from tensorflow.keras.optimizers import Adam
from keras import backend as K
from tensorflow.keras.callbacks import ModelCheckpoint
import json

embed_len = 50
lstm_out = 256

model2 = Sequential([
                    Embedding(input_dim=len(tokenizer.word_index)+1, output_
                              input_length=seq_length),
                    LSTM(lstm_out, return_sequences=True),
                    LSTM(lstm_out),
                    Dense(len(tokenizer.word_index)+1, activation="softmax")
                ])

filepath = "mi_modelo_checkpoint_150.hdf5"

checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1, save_best_
```

```
model2.compile(optimizer=Adam(learning_rate=0.0005), loss="sparse_categorica
history2 = model2.fit(X_train, Y_train, validation_split=0.2, batch_size=102


with open('historial_entrenamiento_150.json', 'w') as f:
    json.dump(history2.history, f)
```

```
Epoch 1/50
984/985 [============================>.] - ETA: 0s - loss: 2.8119
Epoch 1: loss improved from inf to 2.81179, saving model to mi_modelo_checkpo
int_150.hdf5
985/985 [==============================] - 107s 106ms/step - loss: 2.8118 - v
al_loss: 2.3334
Epoch 2/50
985/985 [==============================] - ETA: 0s - loss: 2.1916
Epoch 2: loss improved from 2.81179 to 2.19162, saving model to mi_modelo_che
ckpoint_150.hdf5
985/985 [==============================] - 103s 104ms/step - loss: 2.1916 - v
al_loss: 2.1080
Epoch 3/50
985/985 [==============================] - ETA: 0s - loss: 2.0325
Epoch 3: loss improved from 2.19162 to 2.03249, saving model to mi_modelo_che
ckpoint_150.hdf5
985/985 [==============================] - 103s 104ms/step - loss: 2.0325 - v
al_loss: 2.0063
Epoch 4/50
985/985 [==============================] - ETA: 0s - loss: 1.9606
Epoch 4: loss improved from 2.03249 to 1.96058, saving model to mi_modelo_che
ckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.9606 - v
al_loss: 2.0502
Epoch 5/50
985/985 [==============================] - ETA: 0s - loss: 1.8826
Epoch 5: loss improved from 1.96058 to 1.88259, saving model to mi_modelo_che
ckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.8826 - v
al_loss: 1.8768
Epoch 6/50
985/985 [==============================] - ETA: 0s - loss: 1.8114
Epoch 6: loss improved from 1.88259 to 1.81137, saving model to mi_modelo_che
ckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.8114 - v
al_loss: 1.8333
Epoch 7/50
985/985 [==============================] - ETA: 0s - loss: 1.7749
Epoch 7: loss improved from 1.81137 to 1.77490, saving model to mi_modelo_che
ckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.7749 - v
al_loss: 1.7908
Epoch 8/50
985/985 [==============================] - ETA: 0s - loss: 1.7216
Epoch 8: loss improved from 1.77490 to 1.72161, saving model to mi_modelo_che
ckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.7216 - v
al_loss: 1.7668
Epoch 9/50
985/985 [==============================] - ETA: 0s - loss: 1.6905
Epoch 9: loss improved from 1.72161 to 1.69048, saving model to mi_modelo_che
ckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.6905 - v
al_loss: 1.7328
Epoch 10/50
985/985 [==============================] - ETA: 0s - loss: 1.6395
```

```
Epoch 10: loss improved from 1.69048 to 1.63947, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.6395 - v
al_loss: 1.6931
Epoch 11/50
985/985 [==============================] - ETA: 0s - loss: 1.5961
Epoch 11: loss improved from 1.63947 to 1.59613, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.5961 - v
al_loss: 1.6808
Epoch 12/50
985/985 [==============================] - ETA: 0s - loss: 1.5788
Epoch 12: loss improved from 1.59613 to 1.57878, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.5788 - v
al_loss: 1.6591
Epoch 13/50
985/985 [==============================] - ETA: 0s - loss: 1.5529
Epoch 13: loss improved from 1.57878 to 1.55286, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.5529 - v
al_loss: 1.6468
Epoch 14/50
985/985 [==============================] - ETA: 0s - loss: 1.5279
Epoch 14: loss improved from 1.55286 to 1.52787, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.5279 - v
al_loss: 1.6382
Epoch 15/50
984/985 [============================>.] - ETA: 0s - loss: 1.5006
Epoch 15: loss improved from 1.52787 to 1.50054, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.5005 - v
al_loss: 1.5957
Epoch 16/50
985/985 [==============================] - ETA: 0s - loss: 1.4548
Epoch 16: loss improved from 1.50054 to 1.45483, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.4548 - v
al_loss: 1.5812
Epoch 17/50
985/985 [==============================] - ETA: 0s - loss: 1.4433
Epoch 17: loss improved from 1.45483 to 1.44327, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.4433 - v
al_loss: 1.5858
Epoch 18/50
985/985 [==============================] - ETA: 0s - loss: 1.4824
Epoch 18: loss did not improve from 1.44327
985/985 [==============================] - 103s 104ms/step - loss: 1.4824 - v
al_loss: 1.6164
Epoch 19/50
984/985 [============================>.] - ETA: 0s - loss: 1.4761
Epoch 19: loss did not improve from 1.44327
985/985 [==============================] - 103s 104ms/step - loss: 1.4761 - v
al_loss: 1.5987
```

```
Epoch 20/50
985/985 [==============================] - ETA: 0s - loss: 1.4583
Epoch 20: loss did not improve from 1.44327
985/985 [==============================] - 103s 104ms/step - loss: 1.4583 - v
al_loss: 1.5873
Epoch 21/50
985/985 [==============================] - ETA: 0s - loss: 1.4776
Epoch 21: loss did not improve from 1.44327
985/985 [==============================] - 103s 104ms/step - loss: 1.4776 - v
al_loss: 1.5900
Epoch 22/50
984/985 [============================>.] - ETA: 0s - loss: 1.4649
Epoch 22: loss did not improve from 1.44327
985/985 [==============================] - 102s 104ms/step - loss: 1.4649 - v
al_loss: 1.5793
Epoch 23/50
984/985 [============================>.] - ETA: 0s - loss: 1.4498
Epoch 23: loss did not improve from 1.44327
985/985 [==============================] - 102s 104ms/step - loss: 1.4498 - v
al_loss: 1.5676
Epoch 24/50
984/985 [============================>.] - ETA: 0s - loss: 1.4357
Epoch 24: loss improved from 1.44327 to 1.43571, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.4357 - v
al_loss: 1.5596
Epoch 25/50
985/985 [==============================] - ETA: 0s - loss: 1.4226
Epoch 25: loss improved from 1.43571 to 1.42260, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.4226 - v
al_loss: 1.5495
Epoch 26/50
985/985 [==============================] - ETA: 0s - loss: 1.4101
Epoch 26: loss improved from 1.42260 to 1.41013, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.4101 - v
al_loss: 1.5417
Epoch 27/50
985/985 [==============================] - ETA: 0s - loss: 1.3984
Epoch 27: loss improved from 1.41013 to 1.39838, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3984 - v
al_loss: 1.5338
Epoch 28/50
985/985 [==============================] - ETA: 0s - loss: 1.3873
Epoch 28: loss improved from 1.39838 to 1.38735, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3873 - v
al_loss: 1.5269
Epoch 29/50
985/985 [==============================] - ETA: 0s - loss: 1.3772
Epoch 29: loss improved from 1.38735 to 1.37722, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3772 - v
al_loss: 1.5190
```

```
Epoch 30/50
985/985 [==============================] - ETA: 0s - loss: 1.3673
Epoch 30: loss improved from 1.37722 to 1.36731, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3673 - v
al_loss: 1.5146
Epoch 31/50
985/985 [==============================] - ETA: 0s - loss: 1.3580
Epoch 31: loss improved from 1.36731 to 1.35798, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3580 - v
al_loss: 1.5100
Epoch 32/50
984/985 [============================>.] - ETA: 0s - loss: 1.3491
Epoch 32: loss improved from 1.35798 to 1.34904, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3490 - v
al_loss: 1.5047
Epoch 33/50
985/985 [==============================] - ETA: 0s - loss: 1.3409
Epoch 33: loss improved from 1.34904 to 1.34089, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3409 - v
al_loss: 1.5011
Epoch 34/50
985/985 [==============================] - ETA: 0s - loss: 1.3329
Epoch 34: loss improved from 1.34089 to 1.33285, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3329 - v
al_loss: 1.4985
Epoch 35/50
985/985 [==============================] - ETA: 0s - loss: 1.3248
Epoch 35: loss improved from 1.33285 to 1.32477, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3248 - v
al_loss: 1.4923
Epoch 36/50
984/985 [============================>.] - ETA: 0s - loss: 1.3177
Epoch 36: loss improved from 1.32477 to 1.31770, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3177 - v
al_loss: 1.4888
Epoch 37/50
985/985 [==============================] - ETA: 0s - loss: 1.3099
Epoch 37: loss improved from 1.31770 to 1.30986, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.3099 - v
al_loss: 1.4830
Epoch 38/50
985/985 [==============================] - ETA: 0s - loss: 1.3039
Epoch 38: loss improved from 1.30986 to 1.30392, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 104ms/step - loss: 1.3039 - v
al_loss: 1.4828
Epoch 39/50
985/985 [==============================] - ETA: 0s - loss: 1.2968
```

```
Epoch 39: loss improved from 1.30392 to 1.29679, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 105ms/step - loss: 1.2968 - v
al_loss: 1.4800
Epoch 40/50
985/985 [==============================] - ETA: 0s - loss: 1.2907
Epoch 40: loss improved from 1.29679 to 1.29067, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 104ms/step - loss: 1.2907 - v
al_loss: 1.4765
Epoch 41/50
985/985 [==============================] - ETA: 0s - loss: 1.2848
Epoch 41: loss improved from 1.29067 to 1.28480, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.2848 - v
al_loss: 1.4713
Epoch 42/50
985/985 [==============================] - ETA: 0s - loss: 1.2785
Epoch 42: loss improved from 1.28480 to 1.27845, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 104ms/step - loss: 1.2785 - v
al_loss: 1.4731
Epoch 43/50
985/985 [==============================] - ETA: 0s - loss: 1.2729
Epoch 43: loss improved from 1.27845 to 1.27293, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 104ms/step - loss: 1.2729 - v
al_loss: 1.4694
Epoch 44/50
985/985 [==============================] - ETA: 0s - loss: 1.2693
Epoch 44: loss improved from 1.27293 to 1.26928, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 104ms/step - loss: 1.2693 - v
al_loss: 1.4675
Epoch 45/50
985/985 [==============================] - ETA: 0s - loss: 1.2632
Epoch 45: loss improved from 1.26928 to 1.26321, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.2632 - v
al_loss: 1.4668
Epoch 46/50
985/985 [==============================] - ETA: 0s - loss: 1.2574
Epoch 46: loss improved from 1.26321 to 1.25735, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.2574 - v
al_loss: 1.4590
Epoch 47/50
985/985 [==============================] - ETA: 0s - loss: 1.2521
Epoch 47: loss improved from 1.25735 to 1.25209, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.2521 - v
al_loss: 1.4593
Epoch 48/50
985/985 [==============================] - ETA: 0s - loss: 1.2472
Epoch 48: loss improved from 1.25209 to 1.24716, saving model to mi_modelo_ch
eckpoint_150.hdf5
```

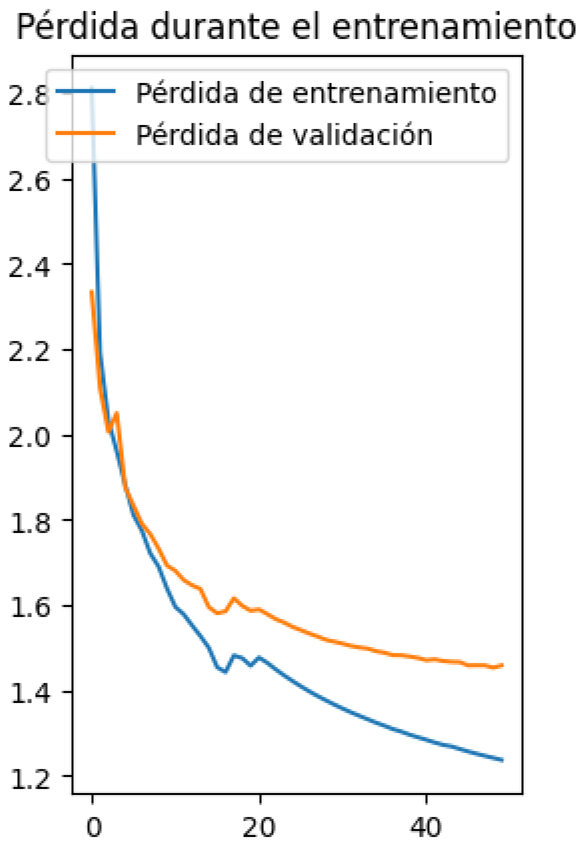
```
985/985 [==============================] - 102s 104ms/step - loss: 1.2472 - v
al_loss: 1.4591
Epoch 49/50
985/985 [==============================] - ETA: 0s - loss: 1.2424
Epoch 49: loss improved from 1.24716 to 1.24241, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 103s 104ms/step - loss: 1.2424 - v
al_loss: 1.4537
Epoch 50/50
985/985 [==============================] - ETA: 0s - loss: 1.2376
Epoch 50: loss improved from 1.24241 to 1.23756, saving model to mi_modelo_ch
eckpoint_150.hdf5
985/985 [==============================] - 102s 104ms/step - loss: 1.2376 - v
al_loss: 1.4590
```

Se grafica la pérdida. A diferencia del primer modelo, se puede observar que la pérdida no llega al menor valor pero sí baja bastante y por parte del set de validación no sube y se mantiene, con más epochs se podrá ver como se quedará estable en un valor. El modelo se comporta bien donde no hay indicios de overfitting.

In [ ]:
```
import matplotlib.pyplot as plt

plt.subplot(1, 2, 2)
plt.plot(history2.history['loss'], label='Pérdida de entrenamiento')
plt.plot(history2.history['val_loss'], label='Pérdida de validación')
plt.title('Pérdida durante el entrenamiento')
plt.legend()

plt.show()
```



In [ ]:
```
import random

random.seed(123)
idx = random.randint(0, len(X_train))
pattern = X_train[idx].flatten().tolist()

print("Initial Pattern : {}".format("".join([tokenizer.index_word[idx] for i

generated_text = []
for i in range(100):
    X_batch = np.array(pattern, dtype=np.int32).reshape(1, seq_length)
    preds = model2.predict(X_batch)
    predicted_index = preds.argmax(axis=-1)[0]
    generated_text.append(predicted_index)
    pattern.append(predicted_index)
    pattern = pattern[1:]

print("Generated Text : {}".format("".join([tokenizer.index_word[idx] for id
```

```
Initial Pattern :  home port of pola , in present @-@ day croatia , except fo
r four engagements . in 1914 , she formed part of the austro @-@ hungarian fl
otilla sent to
1/1 [==============================] - 1s 591ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
```

```
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
Generated Text :  the state of the season .
eceived the second state of the second control of the season .
eceived
```

El resultado se puede ver mucho mejor al anterior, a pesar de que sigue repetiendo las palabras, se pueden ver mucho mejor escritas sin faltas de ortografía.

AdaMax

Al primer modelo se quiso unicamente cambiar el optimizador, en lugar de usar Ada, usar Adamax para poder ver el desempeño y se usó con una longitud de 100 caracteres.

In [ ]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Embedding, Dropout
from tensorflow.keras.optimizers import Adamax
from keras import backend as K
from tensorflow.keras.callbacks import ModelCheckpoint
import json

embed_len = 50
lstm_out = 256

model3 = Sequential([
                    Embedding(input_dim=len(tokenizer.word_index)+1, output_
                              input_length=seq_length),
                    LSTM(lstm_out, return_sequences=True),
                    LSTM(lstm_out),
                    Dense(len(tokenizer.word_index)+1, activation="softmax")
                ])

filepath = "mi_modelo_checkpoint_AdaMax.hdf5"

checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1, save_best_

model3.compile(optimizer=Adamax(learning_rate=0.001), loss="sparse_categoric
history3 = model3.fit(X_train, Y_train, validation_split=0.2, batch_size=102

with open('historial_entrenamiento_AdaMax.json', 'w') as f:
    json.dump(history3.history, f)
```

```
Epoch 1/50
1077/1077 [==============================] - ETA: 0s - loss: 2.9553
Epoch 1: loss improved from inf to 2.95531, saving model to mi_modelo_checkpo
int_AdaMax.hdf5
1077/1077 [==============================] - 82s 74ms/step - loss: 2.9553 - v
al_loss: 2.5562
Epoch 2/50
   2/1077 [..............................] - ETA: 1:09 - loss: 2.5847
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: Us
erWarning: You are saving your model as an HDF5 file via `model.save()`. This
file format is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

```
1076/1077 [============================>.] - ETA: 0s - loss: 2.3685
Epoch 2: loss improved from 2.95531 to 2.36840, saving model to mi_modelo_che
ckpoint_AdaMax.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 2.3684 - v
al_loss: 2.2516
Epoch 3/50
1077/1077 [==============================] - ETA: 0s - loss: 2.1589
Epoch 3: loss improved from 2.36840 to 2.15888, saving model to mi_modelo_che
ckpoint_AdaMax.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 2.1589 - v
al_loss: 2.1059
Epoch 4/50
1076/1077 [============================>.] - ETA: 0s - loss: 2.0379
Epoch 4: loss improved from 2.15888 to 2.03792, saving model to mi_modelo_che
ckpoint_AdaMax.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 2.0379 - v
al_loss: 2.0155
Epoch 5/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.9522
Epoch 5: loss improved from 2.03792 to 1.95222, saving model to mi_modelo_che
ckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.9522 - v
al_loss: 1.9495
Epoch 6/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.8864
Epoch 6: loss improved from 1.95222 to 1.88636, saving model to mi_modelo_che
ckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.8864 - v
al_loss: 1.8953
Epoch 7/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.8331
Epoch 7: loss improved from 1.88636 to 1.83304, saving model to mi_modelo_che
ckpoint_AdaMax.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.8330 - v
al_loss: 1.8521
Epoch 8/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.7880
Epoch 8: loss improved from 1.83304 to 1.78803, saving model to mi_modelo_che
ckpoint_AdaMax.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.7880 - v
al_loss: 1.8189
Epoch 9/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.7494
Epoch 9: loss improved from 1.78803 to 1.74942, saving model to mi_modelo_che
ckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.7494 - v
al_loss: 1.7882
Epoch 10/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.7154
Epoch 10: loss improved from 1.74942 to 1.71533, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.7153 - v
al_loss: 1.7595
Epoch 11/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.6848
Epoch 11: loss improved from 1.71533 to 1.68489, saving model to mi_modelo_ch
```

```
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 77s 71ms/step - loss: 1.6849 - v
al_loss: 1.7375
Epoch 12/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.6576
Epoch 12: loss improved from 1.68489 to 1.65755, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.6576 - v
al_loss: 1.7170
Epoch 13/50
1077/1077 [==============================] - ETA: 0s - loss: 1.6330
Epoch 13: loss improved from 1.65755 to 1.63303, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.6330 - v
al_loss: 1.6972
Epoch 14/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.6103
Epoch 14: loss improved from 1.63303 to 1.61033, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.6103 - v
al_loss: 1.6793
Epoch 15/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5896
Epoch 15: loss improved from 1.61033 to 1.58959, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.5896 - v
al_loss: 1.6668
Epoch 16/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5704
Epoch 16: loss improved from 1.58959 to 1.57042, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.5704 - v
al_loss: 1.6515
Epoch 17/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5526
Epoch 17: loss improved from 1.57042 to 1.55263, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.5526 - v
al_loss: 1.6391
Epoch 18/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5363
Epoch 18: loss improved from 1.55263 to 1.53630, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.5363 - v
al_loss: 1.6283
Epoch 19/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.5214
Epoch 19: loss improved from 1.53630 to 1.52144, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.5214 - v
al_loss: 1.6167
Epoch 20/50
1077/1077 [==============================] - ETA: 0s - loss: 1.5074
Epoch 20: loss improved from 1.52144 to 1.50735, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.5074 - v
```

```
al_loss: 1.6032
Epoch 21/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4943
Epoch 21: loss improved from 1.50735 to 1.49436, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.4944 - v
al_loss: 1.5960
Epoch 22/50
1077/1077 [==============================] - ETA: 0s - loss: 1.4819
Epoch 22: loss improved from 1.49436 to 1.48194, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.4819 - v
al_loss: 1.5894
Epoch 23/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4704
Epoch 23: loss improved from 1.48194 to 1.47037, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.4704 - v
al_loss: 1.5776
Epoch 24/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4595
Epoch 24: loss improved from 1.47037 to 1.45948, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.4595 - v
al_loss: 1.5724
Epoch 25/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4494
Epoch 25: loss improved from 1.45948 to 1.44943, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.4494 - v
al_loss: 1.5638
Epoch 26/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4398
Epoch 26: loss improved from 1.44943 to 1.43980, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.4398 - v
al_loss: 1.5582
Epoch 27/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4306
Epoch 27: loss improved from 1.43980 to 1.43061, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.4306 - v
al_loss: 1.5547
Epoch 28/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4219
Epoch 28: loss improved from 1.43061 to 1.42189, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.4219 - v
al_loss: 1.5492
Epoch 29/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.4142
Epoch 29: loss improved from 1.42189 to 1.41418, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.4142 - v
al_loss: 1.5394
Epoch 30/50
```

```
1076/1077 [============================>.] - ETA: 0s - loss: 1.4061
Epoch 30: loss improved from 1.41418 to 1.40612, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.4061 - v
al_loss: 1.5361
Epoch 31/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3988
Epoch 31: loss improved from 1.40612 to 1.39880, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3988 - v
al_loss: 1.5339
Epoch 32/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3913
Epoch 32: loss improved from 1.39880 to 1.39139, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3914 - v
al_loss: 1.5273
Epoch 33/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3846
Epoch 33: loss improved from 1.39139 to 1.38451, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3845 - v
al_loss: 1.5212
Epoch 34/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3781
Epoch 34: loss improved from 1.38451 to 1.37812, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3781 - v
al_loss: 1.5155
Epoch 35/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3717
Epoch 35: loss improved from 1.37812 to 1.37174, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3717 - v
al_loss: 1.5206
Epoch 36/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3656
Epoch 36: loss improved from 1.37174 to 1.36564, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3656 - v
al_loss: 1.5084
Epoch 37/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3596
Epoch 37: loss improved from 1.36564 to 1.35961, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3596 - v
al_loss: 1.5086
Epoch 38/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3541
Epoch 38: loss improved from 1.35961 to 1.35411, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3541 - v
al_loss: 1.5009
Epoch 39/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3485
Epoch 39: loss improved from 1.35411 to 1.34853, saving model to mi_modelo_ch
```

```
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3485 - v
al_loss: 1.4995
Epoch 40/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3432
Epoch 40: loss improved from 1.34853 to 1.34326, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3433 - v
al_loss: 1.4951
Epoch 41/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3384
Epoch 41: loss improved from 1.34326 to 1.33829, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3383 - v
al_loss: 1.4938
Epoch 42/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3333
Epoch 42: loss improved from 1.33829 to 1.33331, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3333 - v
al_loss: 1.4905
Epoch 43/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3287
Epoch 43: loss improved from 1.33331 to 1.32871, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3287 - v
al_loss: 1.4898
Epoch 44/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3238
Epoch 44: loss improved from 1.32871 to 1.32384, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3238 - v
al_loss: 1.4874
Epoch 45/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3193
Epoch 45: loss improved from 1.32384 to 1.31928, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3193 - v
al_loss: 1.4845
Epoch 46/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3148
Epoch 46: loss improved from 1.31928 to 1.31483, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3148 - v
al_loss: 1.4813
Epoch 47/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3108
Epoch 47: loss improved from 1.31483 to 1.31080, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3108 - v
al_loss: 1.4798
Epoch 48/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3065
Epoch 48: loss improved from 1.31080 to 1.30651, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3065 - v
```

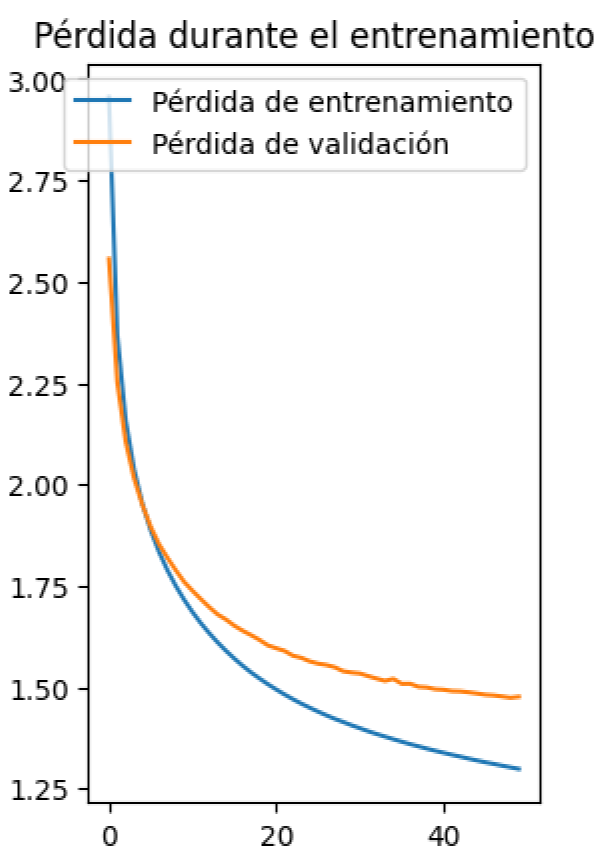
```
al_loss: 1.4771
Epoch 49/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.3024
Epoch 49: loss improved from 1.30651 to 1.30239, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.3024 - v
al_loss: 1.4740
Epoch 50/50
1076/1077 [============================>.] - ETA: 0s - loss: 1.2984
Epoch 50: loss improved from 1.30239 to 1.29840, saving model to mi_modelo_ch
eckpoint_AdaMax.hdf5
1077/1077 [==============================] - 76s 71ms/step - loss: 1.2984 - v
al_loss: 1.4765
```

Se grafica la pérdida del modelo, se puede observar la gráfica que se ve muy suave, no hay mucho ruido. La pérdida baja casi hasta 1.25, muy similar al modelo inicial y el set de validación no se aleja mucho lo que indica que el modelo también se entrena bien y no sufre de overfitting.

In [ ]:
```
import matplotlib.pyplot as plt

plt.subplot(1, 2, 2)
plt.plot(history3.history['loss'], label='Pérdida de entrenamiento')
plt.plot(history3.history['val_loss'], label='Pérdida de validación')
plt.title('Pérdida durante el entrenamiento')
plt.legend()

plt.show()
```



In [ ]:
```
import random

random.seed(123)
idx = random.randint(0, len(X_train))
pattern = X_train[idx].flatten().tolist()

print("Initial Pattern : {}".format("".join([tokenizer.index_word[idx] for i

generated_text = []
for i in range(100):
    X_batch = np.array(pattern, dtype=np.int32).reshape(1, seq_length)
    preds = model3.predict(X_batch)
    predicted_index = preds.argmax(axis=-1)[0]
    generated_text.append(predicted_index)
    pattern.append(predicted_index)
    pattern = pattern[1:]

print("Generated Text : {}".format("".join([tokenizer.index_word[idx] for ic
```

```
Initial Pattern : 1987 – 88 season where he was named the ihl 's co @–@ rooki
e of the year and most valuable player af
1/1 [==============================] - 1s 1s/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
```

```
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
Generated Text : ter the state of the state of the state of the state of the
state of the state of the state of the s
```

Aquí ya empieza unicamente a memorizar la palabra state, lo que puede significar que se está sobreentrenando con los modelos anteriores.

200 palabras y 80 de embedding

Por último se quiere hacer un modelo que pueda entrenarse a partir de 200 palabras y con una longitud de embedding de 80 caracteres.

In [ ]:
```
%%time

import numpy as np
train_dataset, valid_dataset, test_dataset = torchtext.datasets.WikiText2()

seq_length = 200
X_train, Y_train = [], []

for text in X_train_text[:6000]:
    for i in range(0, len(text)-seq_length):
        inp_seq = text[i:i+seq_length].lower()
        out_seq = text[i+seq_length].lower()
        X_train.append(inp_seq)
        Y_train.append(tokenizer.word_index[out_seq])

X_train = tokenizer.texts_to_sequences(X_train)

X_train, Y_train = np.array(X_train, dtype=np.int32), np.array(Y_train)

X_train.shape, Y_train.shape
```

```
CPU times: user 45.5 s, sys: 1.11 s, total: 46.6 s
Wall time: 46.5 s
```

Out[ ]:
```
((1148042, 200), (1148042,))
```

Como se mencionó, se mantiene la estructura original pero ahora se cambia la longitud de embedding.

In [ ]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Embedding, Dropout
from tensorflow.keras.optimizers import Adam
from keras import backend as K
from tensorflow.keras.callbacks import ModelCheckpoint
import json

embed_len = 80
lstm_out = 256

model4 = Sequential([
                    Embedding(input_dim=len(tokenizer.word_index)+1, output_
                              input_length=seq_length),
                    LSTM(lstm_out, return_sequences=True),
                    LSTM(lstm_out),
                    Dense(len(tokenizer.word_index)+1, activation="softmax")
                ])

filepath = "mi_modelo_checkpoint_200.hdf5"

checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1, save_best_

model4.compile(optimizer=Adam(learning_rate=0.001), loss="sparse_categorical
```

```
history4 = model4.fit(X_train, Y_train, validation_split=0.2, batch_size=102

with open('historial_entrenamiento_200.json', 'w') as f:
    json.dump(history4.history, f)
```

```
Epoch 1/50
897/897 [==============================] - ETA: 0s - loss: 2.5499
Epoch 1: loss improved from inf to 2.54992, saving model to mi_modelo_checkpo
int_200.hdf5
897/897 [==============================] - 130s 142ms/step - loss: 2.5499 - v
al_loss: 2.1283
Epoch 2/50
  1/897 [..............................] - ETA: 1:54 - loss: 2.1210
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: Us
erWarning: You are saving your model as an HDF5 file via `model.save()`. This
file format is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

```
897/897 [==============================] - ETA: 0s - loss: 2.0441
Epoch 2: loss improved from 2.54992 to 2.04410, saving model to mi_modelo_che
ckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 2.0441 - v
al_loss: 1.9755
Epoch 3/50
897/897 [==============================] - ETA: 0s - loss: 1.8248
Epoch 3: loss improved from 2.04410 to 1.82485, saving model to mi_modelo_che
ckpoint_200.hdf5
897/897 [==============================] - 128s 142ms/step - loss: 1.8248 - v
al_loss: 1.7785
Epoch 4/50
897/897 [==============================] - ETA: 0s - loss: 1.6438
Epoch 4: loss improved from 1.82485 to 1.64384, saving model to mi_modelo_che
ckpoint_200.hdf5
897/897 [==============================] - 128s 142ms/step - loss: 1.6438 - v
al_loss: 1.6575
Epoch 5/50
897/897 [==============================] - ETA: 0s - loss: 1.6122
Epoch 5: loss improved from 1.64384 to 1.61221, saving model to mi_modelo_che
ckpoint_200.hdf5
897/897 [==============================] - 128s 142ms/step - loss: 1.6122 - v
al_loss: 1.7463
Epoch 6/50
897/897 [==============================] - ETA: 0s - loss: 1.6390
Epoch 6: loss did not improve from 1.61221
897/897 [==============================] - 127s 142ms/step - loss: 1.6390 - v
al_loss: 1.6900
Epoch 7/50
897/897 [==============================] - ETA: 0s - loss: 1.5870
Epoch 7: loss improved from 1.61221 to 1.58695, saving model to mi_modelo_che
ckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.5870 - v
al_loss: 1.6784
Epoch 8/50
897/897 [==============================] - ETA: 0s - loss: 1.5532
Epoch 8: loss improved from 1.58695 to 1.55319, saving model to mi_modelo_che
ckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.5532 - v
al_loss: 1.6310
Epoch 9/50
897/897 [==============================] - ETA: 0s - loss: 1.4939
Epoch 9: loss improved from 1.55319 to 1.49391, saving model to mi_modelo_che
ckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.4939 - v
al_loss: 1.5976
Epoch 10/50
897/897 [==============================] - ETA: 0s - loss: 1.4608
Epoch 10: loss improved from 1.49391 to 1.46083, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.4608 - v
al_loss: 1.5837
Epoch 11/50
897/897 [==============================] - ETA: 0s - loss: 1.4319
Epoch 11: loss improved from 1.46083 to 1.43194, saving model to mi_modelo_ch
eckpoint_200.hdf5
```

```
897/897 [==============================] - 127s 142ms/step - loss: 1.4319 - v
al_loss: 1.5711
Epoch 12/50
897/897 [==============================] - ETA: 0s - loss: 1.4120
Epoch 12: loss improved from 1.43194 to 1.41196, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.4120 - v
al_loss: 1.5611
Epoch 13/50
897/897 [==============================] - ETA: 0s - loss: 1.3919
Epoch 13: loss improved from 1.41196 to 1.39194, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.3919 - v
al_loss: 1.5457
Epoch 14/50
897/897 [==============================] - ETA: 0s - loss: 1.3813
Epoch 14: loss improved from 1.39194 to 1.38131, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.3813 - v
al_loss: 1.5394
Epoch 15/50
897/897 [==============================] - ETA: 0s - loss: 1.3749
Epoch 15: loss improved from 1.38131 to 1.37495, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.3749 - v
al_loss: 1.5363
Epoch 16/50
897/897 [==============================] - ETA: 0s - loss: 1.3665
Epoch 16: loss improved from 1.37495 to 1.36652, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.3665 - v
al_loss: 1.5228
Epoch 17/50
897/897 [==============================] - ETA: 0s - loss: 1.3581
Epoch 17: loss improved from 1.36652 to 1.35807, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.3581 - v
al_loss: 1.5164
Epoch 18/50
897/897 [==============================] - ETA: 0s - loss: 1.3399
Epoch 18: loss improved from 1.35807 to 1.33991, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.3399 - v
al_loss: 1.5102
Epoch 19/50
897/897 [==============================] - ETA: 0s - loss: 1.3270
Epoch 19: loss improved from 1.33991 to 1.32699, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.3270 - v
al_loss: 1.5119
Epoch 20/50
897/897 [==============================] - ETA: 0s - loss: 1.3115
Epoch 20: loss improved from 1.32699 to 1.31152, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.3115 - v
al_loss: 1.4974
```

```
Epoch 21/50
897/897 [==============================] - ETA: 0s - loss: 1.2969
Epoch 21: loss improved from 1.31152 to 1.29690, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.2969 - v
al_loss: 1.4993
Epoch 22/50
897/897 [==============================] - ETA: 0s - loss: 1.2847
Epoch 22: loss improved from 1.29690 to 1.28466, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.2847 - v
al_loss: 1.4942
Epoch 23/50
897/897 [==============================] - ETA: 0s - loss: 1.2752
Epoch 23: loss improved from 1.28466 to 1.27524, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.2752 - v
al_loss: 1.4876
Epoch 24/50
897/897 [==============================] - ETA: 0s - loss: 1.2668
Epoch 24: loss improved from 1.27524 to 1.26679, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.2668 - v
al_loss: 1.4853
Epoch 25/50
897/897 [==============================] - ETA: 0s - loss: 1.2611
Epoch 25: loss improved from 1.26679 to 1.26106, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.2611 - v
al_loss: 1.4672
Epoch 26/50
897/897 [==============================] - ETA: 0s - loss: 1.2578
Epoch 26: loss improved from 1.26106 to 1.25779, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.2578 - v
al_loss: 1.4707
Epoch 27/50
897/897 [==============================] - ETA: 0s - loss: 1.2478
Epoch 27: loss improved from 1.25779 to 1.24785, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.2478 - v
al_loss: 1.4622
Epoch 28/50
897/897 [==============================] - ETA: 0s - loss: 1.2374
Epoch 28: loss improved from 1.24785 to 1.23741, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.2374 - v
al_loss: 1.4682
Epoch 29/50
897/897 [==============================] - ETA: 0s - loss: 1.2270
Epoch 29: loss improved from 1.23741 to 1.22700, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.2270 - v
al_loss: 1.4658
Epoch 30/50
897/897 [==============================] - ETA: 0s - loss: 1.2178
```

```
Epoch 30: loss improved from 1.22700 to 1.21780, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.2178 - v
al_loss: 1.4572
Epoch 31/50
897/897 [==============================] - ETA: 0s - loss: 1.2110
Epoch 31: loss improved from 1.21780 to 1.21105, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.2110 - v
al_loss: 1.4587
Epoch 32/50
897/897 [==============================] - ETA: 0s - loss: 1.2024
Epoch 32: loss improved from 1.21105 to 1.20235, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.2024 - v
al_loss: 1.4577
Epoch 33/50
897/897 [==============================] - ETA: 0s - loss: 1.1940
Epoch 33: loss improved from 1.20235 to 1.19396, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 142ms/step - loss: 1.1940 - v
al_loss: 1.4594
Epoch 34/50
897/897 [==============================] - ETA: 0s - loss: 1.2397
Epoch 34: loss did not improve from 1.19396
897/897 [==============================] - 127s 142ms/step - loss: 1.2397 - v
al_loss: 1.5303
Epoch 35/50
897/897 [==============================] - ETA: 0s - loss: 1.2507
Epoch 35: loss did not improve from 1.19396
897/897 [==============================] - 127s 142ms/step - loss: 1.2507 - v
al_loss: 1.4745
Epoch 36/50
897/897 [==============================] - ETA: 0s - loss: 1.2445
Epoch 36: loss did not improve from 1.19396
897/897 [==============================] - 127s 142ms/step - loss: 1.2445 - v
al_loss: 1.4949
Epoch 37/50
897/897 [==============================] - ETA: 0s - loss: 1.2370
Epoch 37: loss did not improve from 1.19396
897/897 [==============================] - 127s 142ms/step - loss: 1.2370 - v
al_loss: 1.4944
Epoch 38/50
897/897 [==============================] - ETA: 0s - loss: 1.2249
Epoch 38: loss did not improve from 1.19396
897/897 [==============================] - 127s 141ms/step - loss: 1.2249 - v
al_loss: 1.4861
Epoch 39/50
897/897 [==============================] - ETA: 0s - loss: 1.2085
Epoch 39: loss did not improve from 1.19396
897/897 [==============================] - 127s 141ms/step - loss: 1.2085 - v
al_loss: 1.4867
Epoch 40/50
897/897 [==============================] - ETA: 0s - loss: 1.1959
Epoch 40: loss did not improve from 1.19396
897/897 [==============================] - 127s 141ms/step - loss: 1.1959 - v
```

```
al_loss: 1.4818
Epoch 41/50
897/897 [==============================] - ETA: 0s - loss: 1.1886
Epoch 41: loss improved from 1.19396 to 1.18863, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.1886 - v
al_loss: 1.4859
Epoch 42/50
897/897 [==============================] - ETA: 0s - loss: 1.1852
Epoch 42: loss improved from 1.18863 to 1.18523, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.1852 - v
al_loss: 1.4912
Epoch 43/50
897/897 [==============================] - ETA: 0s - loss: 1.1790
Epoch 43: loss improved from 1.18523 to 1.17903, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.1790 - v
al_loss: 1.4944
Epoch 44/50
897/897 [==============================] - ETA: 0s - loss: 1.1698
Epoch 44: loss improved from 1.17903 to 1.16976, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.1698 - v
al_loss: 1.4879
Epoch 45/50
897/897 [==============================] - ETA: 0s - loss: 1.1668
Epoch 45: loss improved from 1.16976 to 1.16676, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.1668 - v
al_loss: 1.4766
Epoch 46/50
897/897 [==============================] - ETA: 0s - loss: 1.1636
Epoch 46: loss improved from 1.16676 to 1.16361, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.1636 - v
al_loss: 1.4631
Epoch 47/50
897/897 [==============================] - ETA: 0s - loss: 1.1565
Epoch 47: loss improved from 1.16361 to 1.15650, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.1565 - v
al_loss: 1.4617
Epoch 48/50
897/897 [==============================] - ETA: 0s - loss: 1.1472
Epoch 48: loss improved from 1.15650 to 1.14720, saving model to mi_modelo_ch
eckpoint_200.hdf5
897/897 [==============================] - 127s 141ms/step - loss: 1.1472 - v
al_loss: 1.4734
Epoch 49/50
897/897 [==============================] - ETA: 0s - loss: 1.1623
Epoch 49: loss did not improve from 1.14720
897/897 [==============================] - 127s 141ms/step - loss: 1.1623 - v
al_loss: 1.5127
Epoch 50/50
897/897 [==============================] - ETA: 0s - loss: 1.1747
```

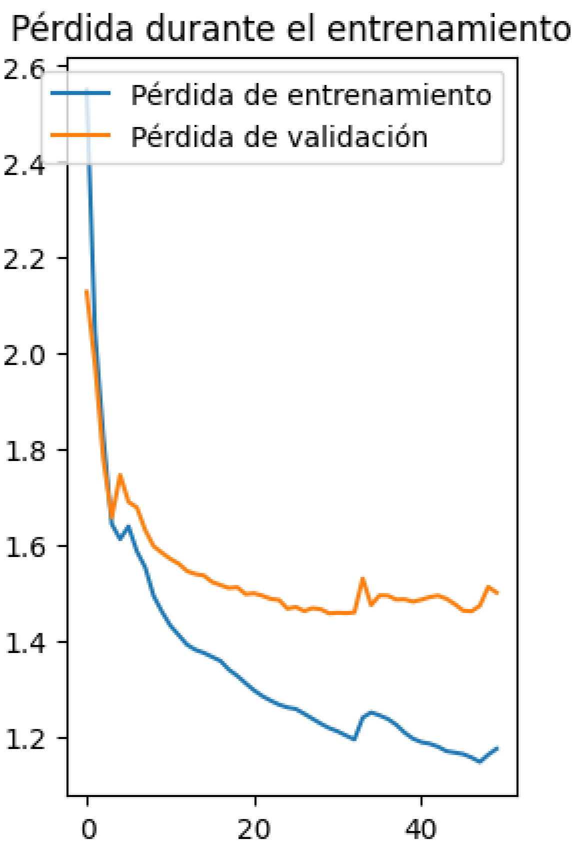
```
Epoch 50: loss did not improve from 1.14720
897/897 [==============================] - 127s 141ms/step - loss: 1.1747 - v
al_loss: 1.5006
```

Se puede observar la gráfica de pérdida en donde se ve mucho ruido, se observa que el loss baja mucho pero al igual que el primer modelo, el set de validación da indicios de overfitting, por lo que no es un buen modelo.

In [ ]:
```python
import matplotlib.pyplot as plt

plt.subplot(1, 2, 2)
plt.plot(history4.history['loss'], label='Pérdida de entrenamiento')
plt.plot(history4.history['val_loss'], label='Pérdida de validación')
plt.title('Pérdida durante el entrenamiento')
plt.legend()

plt.show()
```



In [ ]:
```python
import random

random.seed(123)
idx = random.randint(0, len(X_train))
pattern = X_train[idx].flatten().tolist()

print("Initial Pattern : {}".format("".join([tokenizer.index_word[idx] for i

generated_text = []
for i in range(100):
    X_batch = np.array(pattern, dtype=np.int32).reshape(1, seq_length)
```

```
    preds = model4.predict(X_batch)
    predicted_index = preds.argmax(axis=-1)[0]
    generated_text.append(predicted_index)
    pattern.append(predicted_index)
    pattern = pattern[1:]

print("Generated Text : {}".format("".join([tokenizer.index_word[idx] for ic
```

```
Initial Pattern : es , spreading and <unk> the worship of the old local deiti
es . but others have argued that the most important predynastic gods were , l
ike other elements of egyptian culture , present all across the
1/1 [==============================] - 1s 575ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
```

```
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
Generated Text : state of the series of the series of the series , and the st
ate of the season , and the starts and t
```

Se puede observar cómo se aprende diferentes palabras en su mayoría que comiencen con "s". Se escriben palabras sin mucho sentido pero las escribe bien.

Problemas Encontrados

A lo largo de los diferentes modelos encontrados, se pudo observar que en casi todas las predicciones sí logra generar palabras aunque poco sin sentido. Además, el tiempo de procesamiento para poder completar todo el entrenamiento fue demasiado, por lo que mediante una computadora con mayor procesador puede disminuir el tiempo y dar paso a modelos más complejos o al exploramiento de más parámetros. En todas las gráficas excepto en el modelo que se agregó dropout, se puede ver como no convergen el set de entrenamiento con el de validación pero que sí disminuyen. Por el mismo tiempo de procesamiento no puede ser muy fácil el poder entrenar con muchos caracteres, por lo que se podría mejorar el modelo si se pudiera dar un vector más largo.

Conclusión

Los diferentes modelos utilizados dan paso a un área que ya está bastante explorada y que tiene muchísimas aplicaciones. Los modelos sí mostraron un buen comportamiento en la pérdida, lo que muestra que sí mejora epoch por epoch. A pesar de que las predicciones no son las mejores sí muestran palabras bien escritas sin formar una oración coherente, sí forman frases pero no oraciones coherentes. Para poder mejorar el modelo, es posible hacer diferentes cosas para poder mejorar los modelos.

1. Más epochs para entrenar más tiempo
2. Hacer un modelo más largo para poder mejorar el aprendimiento
3. Conseguir una computadora con mayor procesamiento computacional
4. Probar con otro tipo de redes neuronales
5. Probar diferentes parámetros.

Referencias

Solanki, S. (2022) Keras: Rnns (LSTM) for text generation (character embeddings) by Sunny Solanki, Developed for Developers by Developer for the betterment of Development. Available at: https://coderzcolumn.com/tutorials/artificial-intelligence/keras-text-generation-using-rnn-and-character-embeddings (Accessed: 24 November 2023).

¿Qué son las redes neuronales recurrentes? (no date) IBM. Available at: https://www.ibm.com/mx-es/topics/recurrent-neural-networks (Accessed: 24 November 2023).

English, M.L. in P. (2023) Recurrent neural network-lesson 6: Embeddings and word representations, Medium. Available at: https://medium.com/@nerdjock/recurrent-neural-network-lesson-6-embeddings-and-word-representations-c456f9ce5c69 (Accessed: 24 November 2023).

Team, K. (no date) Keras Documentation: Adam, Keras. Available at: https://keras.io/api/optimizers/adam/ (Accessed: 24 November 2023).

Chollet, F. (2017). Deep learning with python. Manning Publications.

In [ ]: