UNIVERSIDAD DE SANTIAGO DE CHILE

INGENIERÍA ESTADÍSTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Deep Learning

Codificación CIUO 08 bajo redes neuronales

Andrés Rojas Elgueta
Bayron Espinoza Venegas
Bastián Díaz Vergara

Agosto, 2024

Contenido de la presentación

Introducción

- ¿Qué es el CIUO-08?
- Este proyecto presenta la siguiente metodología.

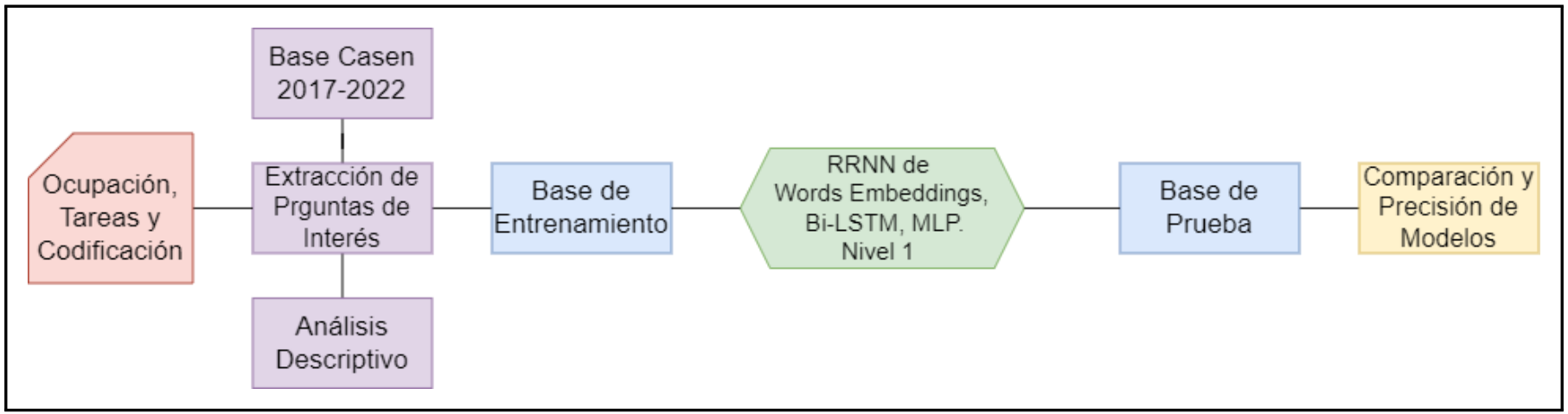

Figura 1: Esquema Metodológico

Resultados clasificador INE

UNIVERSIDAD DE SANTIAGO DE CHILE

Cuadro 11: Resultados CIUO 1 dígito

modelo	acc	macro	micro	weighted
secuencias feed-forward 1d	0.8858	0.8599	0.8858	0.8855
TF-IDF feed-forward 1d	0.8684	0.8362	0.8684	0.8686
embeddings feed-forward 1d	0.8793	0.8519	0.8793	0.8807
embeddings Gated Recurrent Unit 1d	0.8989	0.8796	0.8989	0.8990

Fuente: elaboración propia, Instituto Nacional de Estadísticas.

Cuadro 12: Resultados CIUO 2 dígitos

modelo	acc	macro	micro	weighted
secuencias feed-forward 2d	0.8456	0.7249	0.8456	0.8476
TF-IDF feed-forward 2d	0.8412	0.7355	0.8412	0.8431
embeddings feed-forward 2d	0.8324	0.7220	0.8324	0.8348
embeddings Gated Recurrent Unit 2d	0.8526	0.7364	0.8526	0.8543

Fuente: elaboración propia, Instituto Nacional de Estadísticas.

Objetivos

Objetivo General: Desarrollar un modelo de aprendizaje profundo para la codificación de al menos un nivel de la Clasificación Internacional Uniforme de Ocupaciones 2008 (CIUO-08).

Objetivos Específicos

- Representar de manera vectorial las respuestas textuales, capturando la relación semántica de las respuestas mediante la técnica de Words Embeddings.
- Implementar una red neuronal recurrente tipo LSTM bidireccional con MLP para codificar cada nivel de la clasificación a partir de la representación vectorial.

Descripción de Datos

Nombre	Descripción	Tipo	Valor
Ocupación	Ocupación del Encuestado	Texto	Hasta 249 Caracteres
Tarea	Tareas que cumple en su Ocupación	Texto	Hasta 492 Caracteres
CIUO_N	Clasificación Internacional Uniforme de Ocupación , desde Nivel 1 a Nivel 4	Factor	Nivel 1: 10 Grupos Nivel 2: 44 Grupos Nivel 3: 162 Grupos Nivel 4: 649 Ocupaciones

Tabla 1: Descripción de Variables

Descripción de los datos

Recordemos que las clasificaciones del 1 al 9 corresponde a una tarea u ocupación de la fuerza de trabajo chilena

Código	Tarea y ocupación
1	Directores, gerentes y administradores
2	Profesionales, científicos e intelectuales
3	Técnicos y profesionales de nivel medio
4	Personal de apoyo administrativo
5	Trabajadores de los servicios y vendedores de comercios y mercados
6	Agricultores y trabajadores calificados agropecuarios, forestales y pesqueros
7	Artesanos y operarios de oficios
8	Operadores de instalaciones, máquinas y ensambladores
9	Ocupaciones elementales
0	Ocupaciones de las fuerzas armadas

Tabla 2: Clasificación del primer nivel CIUO-08CL

Proporción por categorías nivel 1

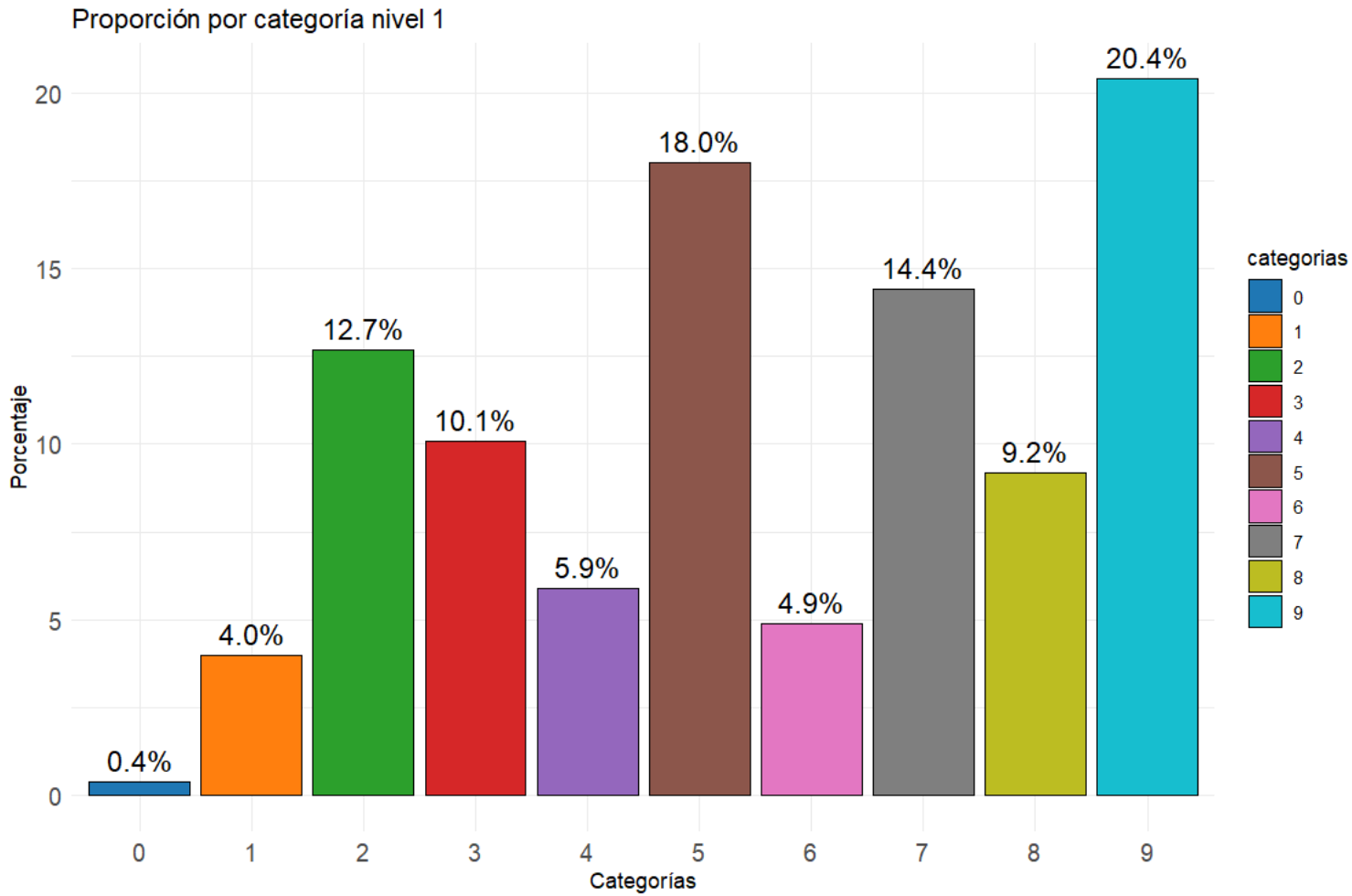

Figura 2: Gráfico de barras porcentual

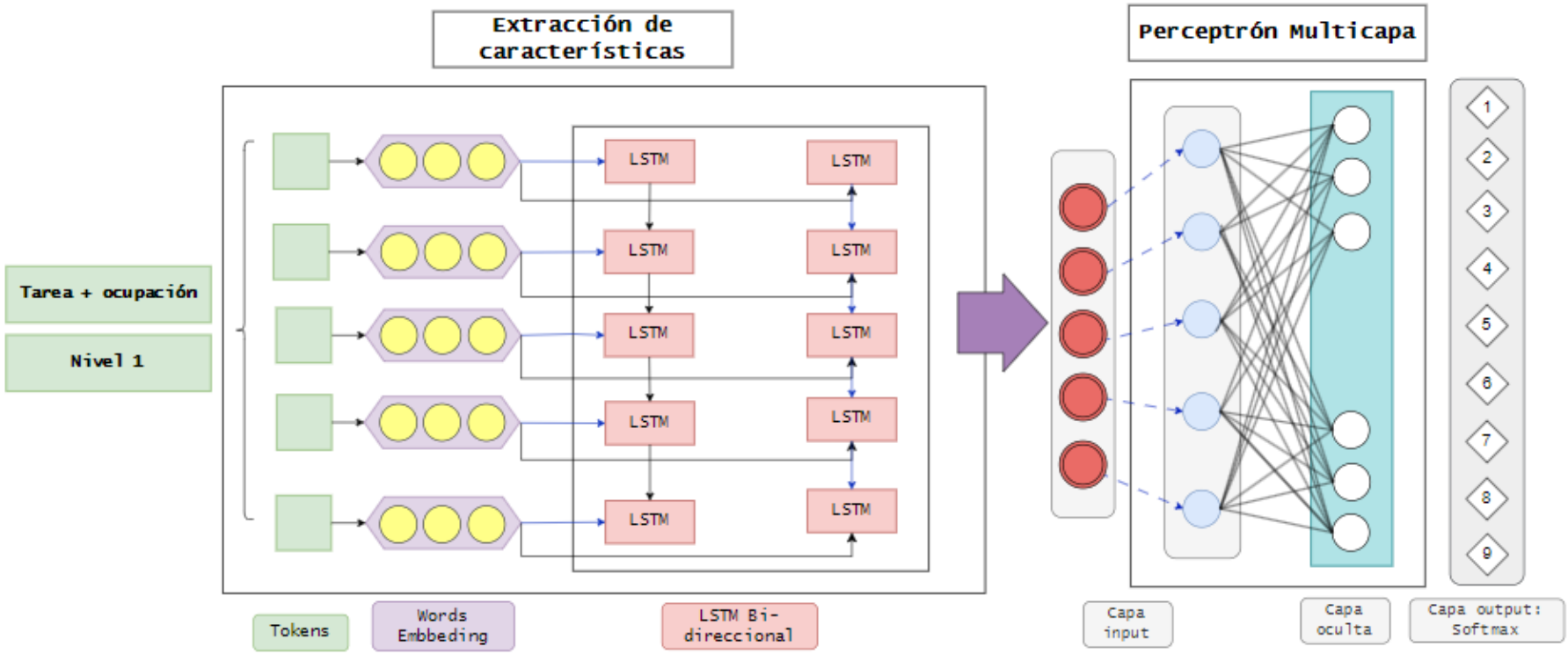

Figura 3: Metodología para el nivel uno de codificación

Descripción de Proceso de Entrenamiento

Descripción de Entrenamiento	
Uso de Datos	70 % Entrenamiento 30 % Validación
Épocas	10 Patience: 3 val_loss Early Stop: 7
Función de Pérdida	Sparse Categorical Crossentropy
GPU	No
Plataforma	R
Tiempo de Computo	1 hora 0 minutos 51 s
Métricas de Validación	Precisión Matriz de Confusión Macro average Micro average Weighted average

Tabla 3: Descripción de Entrenamiento

Capa	Neuronas
Bidirectional LSTM Layer 1	256
Bidirectional LSTM Layer 2	64
Dense Layer 1	16
Dense Layer 2 (Output Layer)	9
Total de neuronas	**345**

Tabla 4: Resumen de capas y neuronas del modelo

Capa	Regularización	Activación
Dense Layer 1	(Lasso(L1) & Ridge(L2))=0.01 Dropout = 0.45	Relu
Dense Layer 2	-	Softmax

Tabla 5: Regularización y función de activación

Descripción del Proceso de Entrenamiento

UNIVERSIDAD DE SANTIAGO DE CHILE

```
### Ajuste del modelo ###
model <- keras_model_sequential() %>%
  layer_embedding(input_dim = vocab_size, output_dim = 128, mask_zero = TRUE)
    ↪ %>%
  bidirectional(layer_lstm(units = 128, return_sequences = TRUE)) %>%
  bidirectional(layer_lstm(units = 32)) %>%
  layer_dense(units = 16, activation = 'relu', kernel_regularizer = regularizer_l1_l2(l1 =
    ↪ 0.01, l2 = 0.01)) %>%
  layer_dropout(rate = 0.45) %>%
  layer_dense(units = num_classes, activation = 'softmax')
### Parada temprana ###
early_stop <- callback_early_stopping(
  monitor = "val_loss",
  patience = 3,
  restore_best_weights = TRUE
)
### Compilación del modelo ###
model %>% compile(
  loss = 'sparse_categorical_crossentropy',
  optimizer = optimizer_adam(learning_rate = 0.001, beta_1 = 0.9),
  metrics = c('accuracy')
)
### Entrenar el modelo ###
tiempo_ini = Sys.time()
history <- model %>% fit(
  train_dataset,      # Conjunto de datos de entrenamiento
  epochs = 10,        # Número de épocas
  validation_data = test_dataset,  # Conjunto de datos de validación
  callbacks = list(early_stop)
)
tiempo_fin = Sys.time()
```

Figura 4: Código

Modelo de RRNN para CIUO-08

```
> summary(model)
Model: "Modelo elegido"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, None, 128)         3112832
 bidirectional_3 (Bidirectional)  (None, None, 256)    263168
 bidirectional_2 (Bidirectional)  (None, 64)           73984
 dense_3 (Dense)             (None, 16)                1040
 dropout_1 (Dropout)         (None, 16)                0
 dense_2 (Dense)             (None, 9)                 153
=================================================================
Total params: 3451177 (13.17 MB)
Trainable params: 3451177 (13.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figura 5: Resumen del modelo previo a entrenamiento

Épocas

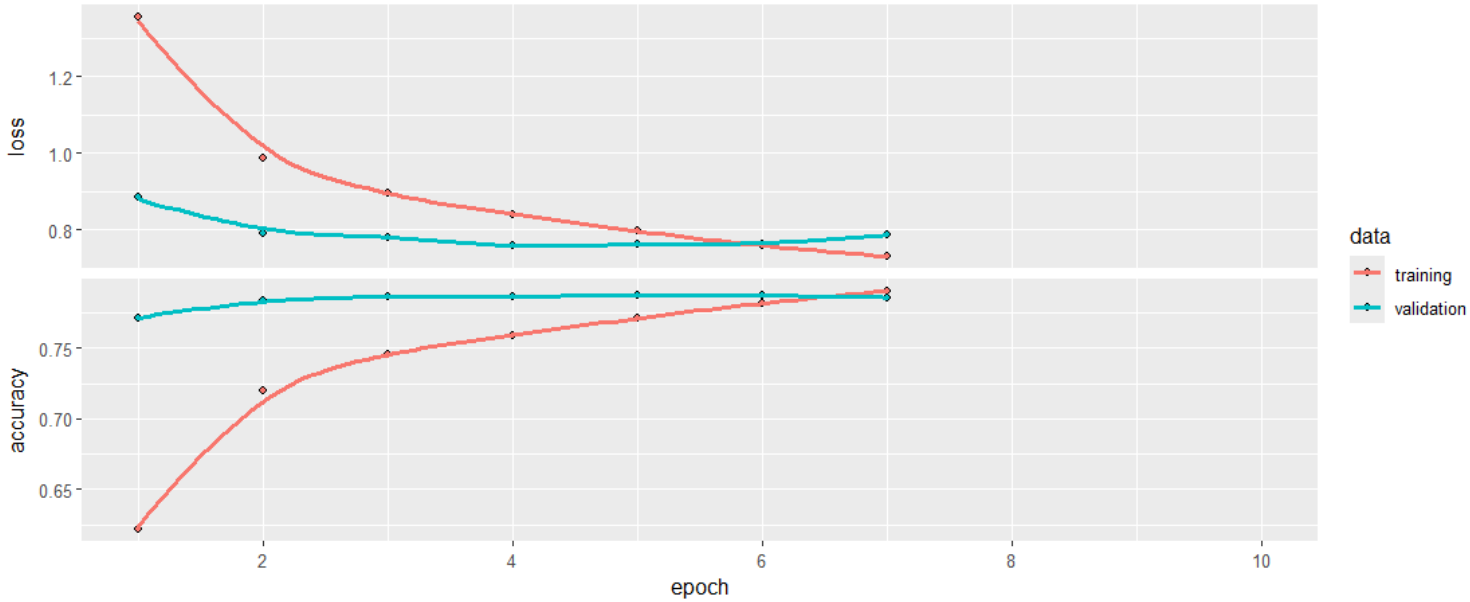

Figura 6: Ajuste por épocas

Época	1	2	3	4	5	6	7
Acc validation	0.771	0.784	0.787	0.787	0.788	0.787	0.786
Acc training	0.621	0.720	0.745	0.759	0.771	0.782	0.791

Época	1	2	3	4	5	6	7
Loss validation	0.884	0.791	0.780	0.758	0.762	0.763	0.786
Loss training	1.355	0.987	0.896	0.840	0.796	0.758	0.731

Validación: Métricas

Predicted / Actual	1	2	3	4	5	6	7	8	9
1	999	80	77	20	164	10	16	5	12
2	250	5772	904	159	97	6	59	16	33
3	164	523	3093	494	360	36	191	66	84
4	28	77	501	2155	99	7	7	45	119
5	536	113	423	168	7882	28	128	49	651
6	48	18	39	16	63	2062	74	42	555
7	57	111	329	18	108	51	6711	232	421
8	13	17	48	43	41	40	167	4317	198
9	13	23	52	59	797	345	315	181	8800

Tabla 6: Matriz de Confusión

Validación: Métricas

Categoría	1	2	3	4	5	6	7	8	9
% Aciertos	47.39 %	85.71 %	56.59 %	68.81 %	82.02 %	79.77 %	87.52 %	87.16 %	80.93 %

Tabla 7: Porcentaje de aciertos por categoría

Métrica	**Fórmula**	**INE**	**Modelo Propuesto**
Precisión	$Acc = \frac{\text{Predicciones Correctas}}{\text{Predicciones Totales}}$	0.8989	0.7876
Macro Average	$Macro = \frac{1}{N}\sum_{i=1}^{N} \text{Presición}_i$	0.8796	0.7509
Micro Average	$Micro = \frac{\sum \text{Verdaderos Positivos}}{\sum \text{Verdaderos Positivos} + \sum \text{Falsos Positivos}}$	0.8989	0.7866
Weighted Average	$Weighted = \frac{\sum_{i=1}^{N}(\text{Tamaño Categoría}_i \cdot \text{Precisión}_i)}{\text{Tamaño Total}}$	0.8990	0.7868

Tabla 8: Métricas del modelo

Conclusiones

- Se logró implementar una red neuronal recurrente tipo LSTM bidireccional con MLP para codificar el primer nivel de clasificación a partir de una representación vectorial mediante words embeddings.
- En los grupos 2, 7 y 8 se logra una presición superior al 85 % y los grupos 5, 6 y 9 alcanzan alrededor del 80 % de presición.
- El grupo 6 a pesar de ser el tercer grupo con menos datos recopilados alcanza alrededor del 80 % de presición.
- La inclusión de capas LSTM bidireccionales no proporciona una mejora en la presición de la codificación a nivel 1 respecto a la arquitectura propuesta por INE.

Trabajos Futuros

- Se cree que entrenar un *words embeddings* propio y agregarlo al flujo de trabajo aumentará significativamente la presición del modelo.
- Se propone implementar algoritmos de *boosting* y *baggins* para mejorar las métricas de validación, inclusive incorporarlas al flujo de trabajo ya propuesto, ya que su naturaleza resulta ideal para clasificar de manera jerárquica.

Referencias

1. Géron, Aurélien. 2022. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc
2. Instituto Nacional de Estadísticas. (2018). Clasificador Chileno de Ocupaciones CIUO 08.CL. Instituto Nacional de Estadísticas, Subdirección Técnica, Departamento de Infraestructura Económica, Sección de Nomenclaturas. Disponible en `https://www.ine.gob.cl/docs/default-source/buenas-practicas/clasificaciones/ciuo/clasificador/ciuo-08-cl.pdf`
3. Gautam, H. (2020, marzo 1). Word Embedding. Basics. `https://medium.com/@hari4om/word-embedding-d816f643140`
4. DataScientest. (s.f.). Memoria a largo plazo a corto plazo (LSTM): ¿Qué es?. DataScientest. (2024, mayo 20) `https://datascientest.com/es/memoria-a-largo-plazo-a-corto-plazo-lstm`