Multiclass Text Classification with

Logistic Regression Implemented with PyTorch and CE Loss

Luis Maximiliano López Ramírez - A00833321

First, we will do some initialization.

En este bloque se importan las librerías necesarias para el proyecto: random, torch, numpy, pandas, y tqdm para manejar datos y mostrar barras de progreso. Se configura si se usará la GPU (si está disponible) para acelerar el procesamiento, y se establece una semilla aleatoria para asegurar que los resultados sean reproducibles cada vez que se ejecute el código.

In [1]:
```
import random
import torch
import numpy as np
import pandas as pd
from tqdm.notebook import tqdm

# enable tqdm in pandas
tqdm.pandas()

# set to True to use the gpu (if there is one available)
use_gpu = True

# select device
device = torch.device('cuda' if use_gpu and torch.cuda.is_available() else 'cpu')
print(f'device: {device.type}')

# random seed
seed = 1234

# set random seed
if seed is not None:
    print(f'random seed: {seed}')
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
```

```
device: cuda
random seed: 1234
```

We will be using the AG's News Topic Classification Dataset. It is stored in two CSV files: `train.csv` and `test.csv`, as well as a `classes.txt` that stores the labels of the

classes to predict.

First, we will load the training dataset using pandas and take a quick look at how the data.

Aquí se carga el conjunto de datos de entrenamiento desde un archivo CSV usando pandas. Se especifica que no hay encabezados (header=None) para luego asignar manualmente nombres a las columnas: 'class index' (índice de la clase), 'title' (título) y 'description' (descripción). Esto organiza los datos y facilita el acceso a las distintas partes del dataset para su procesamiento posterior.

In [2]:
```
train_df = pd.read_csv('train (1).csv', header=None)
train_df.columns = ['class index', 'title', 'description']
train_df
```

Out[2]:

	class index	title	description
0	3	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...
1	3	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...
2	3	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...
3	3	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...
4	3	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...
...	...	...	...
119995	1	Pakistan's Musharraf Says Won't Quit as Army C...	KARACHI (Reuters) - Pakistani President Perve...
119996	2	Renteria signing a top-shelf deal	Red Sox general manager Theo Epstein acknowled...
119997	2	Saban not going to Dolphins yet	The Miami Dolphins will put their courtship of...
119998	2	Today's NFL games	PITTSBURGH at NY GIANTS Time: 1:30 p.m. Line: ...
119999	2	Nets get Carter from Raptors	INDIANAPOLIS -- All-Star Vince Carter was trad...

120000 rows × 3 columns

The dataset consists of 120,000 examples, each consisting of a class index, a title, and a description. The class labels are distributed in a separated file. We will add the labels to the

dataset so that we can interpret the data more easily. Note that the label indexes are one-based, so we need to subtract one to retrieve them from the list.

En este bloque se cargan las etiquetas de las clases desde un archivo classes.txt, donde cada línea representa una clase distinta. Luego, se utiliza la columna 'class index' del DataFrame para mapear cada índice a su correspondiente etiqueta, ajustando los índices porque son uno-basados. Finalmente, se inserta una nueva columna llamada 'class' en el DataFrame, que contiene las etiquetas legibles para facilitar la interpretación y el análisis de los datos.

In [3]:
```
labels = open('classes.txt').read().splitlines()
classes = train_df['class index'].map(lambda i: labels[i-1])
train_df.insert(1, 'class', classes)
train_df
```

Out[3]:

	class index	class	title	description
0	3	Business	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...
1	3	Business	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...
2	3	Business	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...
3	3	Business	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...
4	3	Business	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...
...	...	...	...	...
119995	1	World	Pakistan's Musharraf Says Won't Quit as Army C...	KARACHI (Reuters) - Pakistani President Perve...
119996	2	Sports	Renteria signing a top-shelf deal	Red Sox general manager Theo Epstein acknowled...
119997	2	Sports	Saban not going to Dolphins yet	The Miami Dolphins will put their courtship of...
119998	2	Sports	Today's NFL games	PITTSBURGH at NY GIANTS Time: 1:30 p.m. Line: ...
119999	2	Sports	Nets get Carter from Raptors	INDIANAPOLIS -- All-Star Vince Carter was trad...

120000 rows × 4 columns

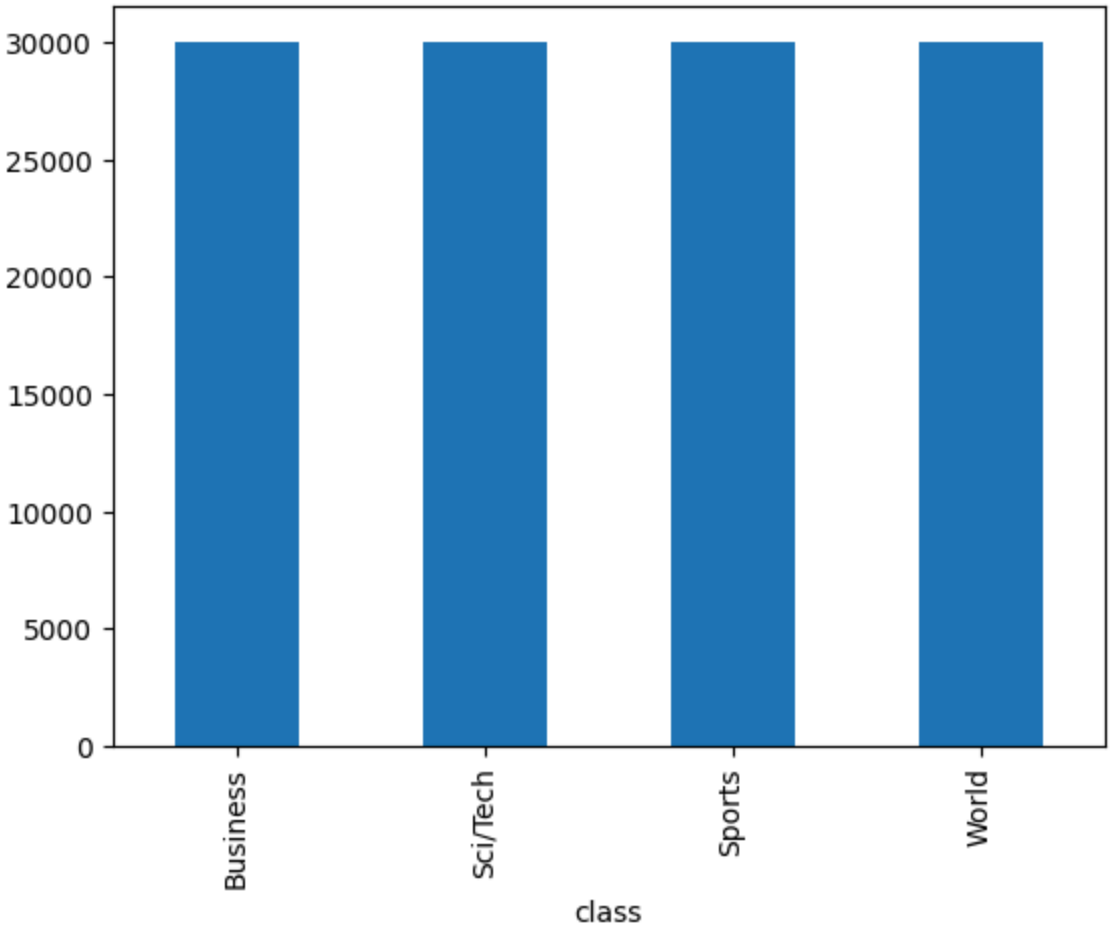
Let's inspect how balanced our examples are by using a bar plot.

Este bloque genera un gráfico de barras que muestra la distribución de las clases en el conjunto de datos de entrenamiento. Utiliza pd.value_counts para contar la cantidad de ejemplos en cada clase, y luego plot.bar() para visualizar cuán equilibradas están las clases. Esto ayuda a identificar si el dataset está balanceado o si hay clases que podrían necesitar un ajuste durante el entrenamiento.

In [4]:
```
pd.value_counts(train_df['class']).plot.bar()
```

```
C:\Users\luism\AppData\Local\Temp\ipykernel_23352\1245903889.py:1: FutureWarning: pa
ndas.value_counts is deprecated and will be removed in a future version. Use pd.Seri
es(obj).value_counts() instead.
  pd.value_counts(train_df['class']).plot.bar()
```

Out[4]: <Axes: xlabel='class'>



The classes are evenly distributed. That's great!

However, the text contains some spurious backslashes in some parts of the text. They are meant to represent newlines in the original text. An example can be seen below, between the words "dwindling" and "band".

Aquí se imprime la descripción del primer registro en el conjunto de datos de entrenamiento. Esto permite inspeccionar de manera rápida el contenido de la columna 'description' y detectar posibles problemas o peculiaridades en el texto, como caracteres no

deseados o formatos inconsistentes que podrían requerir limpieza antes de procesar los datos para el modelo.

In [5]:
```
print(train_df.loc[0, 'description'])
```

```
Reuters - Short-sellers, Wall Street's dwindling\band of ultra-cynics, are seeing gr
een again.
```

We will replace the backslashes with spaces on the whole column using pandas replace method.

En este bloque se limpian y preparan los textos para el modelo. Primero, se convierten las columnas 'title' y 'description' a minúsculas para evitar problemas de diferenciación por mayúsculas/minúsculas. Luego, se combinan ambas columnas en una nueva columna llamada 'text', que contiene el título y la descripción unidos. Finalmente, se reemplazan las barras invertidas (\) con espacios para limpiar el texto de caracteres indeseados. Esto asegura que el texto esté en un formato uniforme y limpio para el procesamiento posterior.

In [6]:
```
title = train_df['title'].str.lower()
descr = train_df['description'].str.lower()
text = title + " " + descr
train_df['text'] = text.str.replace('\\', ' ', regex=False)
train_df
```

Out[6]:

	class index	class	title	description	text
0	3	Business	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...	wall st. bears claw back into the black (reute...
1	3	Business	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...	carlyle looks toward commercial aerospace (reu...
2	3	Business	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...	oil and economy cloud stocks' outlook (reuters...
3	3	Business	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...	iraq halts oil exports from main southern pipe...
4	3	Business	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...	oil prices soar to all-time record, posing new...
...	...	...	...	...	...
119995	1	World	Pakistan's Musharraf Says Won't Quit as Army C...	KARACHI (Reuters) - Pakistani President Perve...	pakistan's musharraf says won't quit as army c...
119996	2	Sports	Renteria signing a top-shelf deal	Red Sox general manager Theo Epstein acknowled...	renteria signing a top-shelf deal red sox gene...
119997	2	Sports	Saban not going to Dolphins yet	The Miami Dolphins will put their courtship of...	saban not going to dolphins yet the miami dolp...
119998	2	Sports	Today's NFL games	PITTSBURGH at NY GIANTS Time: 1:30 p.m. Line: ...	today's nfl games pittsburgh at ny giants time...
119999	2	Sports	Nets get Carter from Raptors	INDIANAPOLIS -- All-Star Vince Carter was trad...	nets get carter from raptors indianapolis -- a...

120000 rows × 5 columns

Now we will proceed to tokenize the title and description columns using NLTK's word_tokenize(). We will add a new column to our dataframe with the list of tokens.

Este bloque descarga el paquete punkt de NLTK, que contiene herramientas para la tokenización de texto. La tokenización es el proceso de dividir un texto en palabras individuales (tokens), lo cual es esencial para el análisis de lenguaje natural. Descargar este

paquete asegura que se puedan usar funciones como word_tokenize() para dividir los textos en palabras de manera precisa.

In [7]:
```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\luism\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[7]: True

Aquí se realiza la tokenización del texto en el conjunto de datos de entrenamiento. Primero, se toma una muestra aleatoria del 55% de los datos usando sample para reducir el tamaño del dataset durante el desarrollo o pruebas *debido a la falta merioria RAM*. Luego, se utiliza word_tokenize de NLTK para dividir el texto de la columna 'text' en palabras individuales (tokens), y se almacena el resultado en una nueva columna llamada 'tokens'. El uso de progress_map de tqdm permite ver el progreso del proceso de tokenización.

In [8]:
```
from nltk.tokenize import word_tokenize

train_df = train_df.sample(frac=0.55, random_state=42)

train_df['tokens'] = train_df['text'].progress_map(word_tokenize)
train_df
```

```
  0%|          | 0/66000 [00:00<?, ?it/s]
```

Out[8]:

	class index	class	title	description	text	tokens
71787	3	Business	BBC set for major shake-up, claims newspaper	London - The British Broadcasting Corporation,...	bbc set for major shake-up, claims newspaper l...	[bbc, set, for, major, shake-up, ,, claims, ne...
67218	3	Business	Marsh averts cash crunch	Embattled insurance broker #39;s banks agree t...	marsh averts cash crunch embattled insurance b...	[marsh, averts, cash, crunch, embattled, insur...
54066	2	Sports	Jeter, Yankees Look to Take Control (AP)	AP - Derek Jeter turned a season that started ...	jeter, yankees look to take control (ap) ap - ...	[jeter, ,, yankees, look, to, take, control, (...
7168	4	Sci/Tech	Flying the Sun to Safety	When the Genesis capsule comes back to Earth w...	flying the sun to safety when the genesis caps...	[flying, the, sun, to, safety, when, the, gene...
29618	3	Business	Stocks Seen Flat as Nortel and Oil Weigh	NEW YORK (Reuters) - U.S. stocks were set to ...	stocks seen flat as nortel and oil weigh new ...	[stocks, seen, flat, as, nortel, and, oil, wei...
...	...	...	...	...	...	...
43622	1	World	Italian hostages #39; release gives hope to Bi...	The brother of British hostage Kenneth Bigley ...	italian hostages #39; release gives hope to bi...	[italian, hostages, #, 39, ;, release, gives, ...
106452	1	World	Do They Know It's Simplistic?	Band Aid's intentions are good, but Africa nee...	do they know it's simplistic? band aid's inten...	[do, they, know, it, 's, simplistic, ?, band, ...
8687	4	Sci/Tech	Google at Bottom of ISS Governance Ranking	Google Inc. tried to democratize the IPO proce...	google at bottom of iss governance ranking goo...	[google, at, bottom, of, iss, governance, rank...
107858	3	Business	Nokia regains share of global mobile handset m...	NEW YORK, December 3 (newratings.com) - Nokia ...	nokia regains share of global mobile handset m...	[nokia, regains, share, of, global, mobile, ha...
50712	3	Business	Kodak #39;s Restructuring Develops	Why is it that so many times people take pictu...	kodak #39;s restructuring develops why is it t...	[kodak, #, 39, ;, s, restructuring, develops, ...

66000 rows × 6 columns

Now we will create a vocabulary from the training data. We will only keep the terms that repeat beyond some threshold established below.

En este bloque se crea un vocabulario a partir de los tokens en el conjunto de datos de entrenamiento. Primero, se establecen como límite los tokens que aparezcan más de 10 veces (threshold = 10) para filtrar palabras poco frecuentes. Luego, se usa explode y value_counts para contar cuántas veces aparece cada token. Se seleccionan solo aquellos que cumplen con el umbral definido. A continuación, se construyen dos estructuras: id_to_token, que es una lista de tokens donde el primer elemento es [UNK] (para representar palabras desconocidas), y token_to_id, que es un diccionario que asigna un identificador único a cada token. Finalmente, se imprime el tamaño del vocabulario, es decir, cuántos tokens distintos se mantienen tras el filtrado. Esto optimiza el modelo al reducir el ruido y la complejidad del texto procesado.

In [9]:
```
threshold = 10
tokens = train_df['tokens'].explode().value_counts()
tokens = tokens[tokens > threshold]
id_to_token = ['[UNK]'] + tokens.index.tolist()
token_to_id = {w:i for i,w in enumerate(id_to_token)}
vocabulary_size = len(id_to_token)
print(f'vocabulary size: {vocabulary_size:,}')
```

```
vocabulary size: 14,226
```

En este bloque se define y utiliza una función para transformar los tokens de cada texto en vectores de características numéricas, lo que es necesario para entrenar el modelo. La función make_feature_vector toma una lista de tokens y crea un diccionario (defaultdict) donde las claves son los identificadores de cada token (usando token_to_id), y los valores son las frecuencias de esos tokens. Si un token no está en el vocabulario, se le asigna el identificador de [UNK]. Luego, se aplica esta función a cada fila del DataFrame para generar una nueva columna llamada 'features', que contiene el vector de características de cada texto. Esto convierte los textos en una forma más manejable para el modelo, ya que ahora se pueden representar como datos numéricos.

In [10]:
```
from collections import defaultdict

def make_feature_vector(tokens, unk_id=0):
    vector = defaultdict(int)
    for t in tokens:
        i = token_to_id.get(t, unk_id)
        vector[i] += 1
    return vector

train_df['features'] = train_df['tokens'].progress_map(make_feature_vector)
train_df
```

```
  0%|          | 0/66000 [00:00<?, ?it/s]
```

Out[10]:

	class index	class	title	description	text	tokens	features
71787	3	Business	BBC set for major shake-up, claims newspaper	London - The British Broadcasting Corporation,...	bbc set for major shake-up, claims newspaper l...	[bbc, set, for, major, shake-up, ,, claims, ne...	{2499: 1, 170: 1, 11: 1, 190: 1, 6093: 2, 2: 5...
67218	3	Business	Marsh averts cash crunch	Embattled insurance broker #39;s banks agree t...	marsh averts cash crunch embattled insurance b...	[marsh, averts, cash, crunch, embattled, insur...	{1904: 2, 0: 4, 737: 1, 4961: 1, 2843: 1, 729:...
54066	2	Sports	Jeter, Yankees Look to Take Control (AP)	AP - Derek Jeter turned a season that started ...	jeter, yankees look to take control (ap) ap - ...	[jeter, ,, yankees, look, to, take, control, (...	{6488: 2, 2: 1, 503: 1, 605: 1, 4: 1, 196: 1, ...
7168	4	Sci/Tech	Flying the Sun to Safety	When the Genesis capsule comes back to Earth w...	flying the sun to safety when the genesis caps...	[flying, the, sun, to, safety, when, the, gene...	{2542: 1, 1: 4, 419: 2, 4: 3, 1043: 1, 96: 1, ...
29618	3	Business	Stocks Seen Flat as Nortel and Oil Weigh	NEW YORK (Reuters) - U.S. stocks were set to ...	stocks seen flat as nortel and oil weigh new ...	[stocks, seen, flat, as, nortel, and, oil, wei...	{161: 2, 621: 1, 1515: 1, 21: 1, 2113: 2, 9: 1...
...	...	...	...	...	...	...	...
43622	1	World	Italian hostages #39; release gives hope to Bi...	The brother of British hostage Kenneth Bigley ...	italian hostages #39; release gives hope to bi...	[italian, hostages, #, 39, ;, release, gives, ...	{831: 2, 1048: 1, 12: 1, 13: 1, 8: 1, 368: 2, ...
106452	1	World	Do They Know It's Simplistic?	Band Aid's intentions are good, but Africa nee...	do they know it's simplistic? band aid's inten...	[do, they, know, it, 's, simplistic, ?, band, ...	{338: 1, 74: 1, 1113: 1, 29: 1, 23: 2, 0: 2, 8...
8687	4	Sci/Tech	Google at Bottom of ISS	Google Inc. tried to democratize the IPO proce...	google at bottom of iss governance	[google, at, bottom, of, iss,	{192: 2, 22: 1, 2566: 1, 6: 1,

	class index	class	title	description	text	tokens	features
			Governance Ranking		ranking goo...	governance, rank...	5744: 1, 5176: ...
107858	3	Business	Nokia regains share of global mobile handset m...	NEW YORK, December 3 (newratings.com) - Nokia ...	nokia regains share of global mobile handset m...	[nokia, regains, share, of, global, mobile, ha...	{1001: 2, 13268: 1, 397: 2, 6: 1, 310: 2, 251:...
50712	3	Business	Kodak #39;s Restructuring Develops	Why is it that so many times people take pictu...	kodak #39;s restructuring develops why is it t...	[kodak, #, 39, ;, s, restructuring, develops, ...	{2586: 1, 12: 2, 13: 2, 8: 2, 17: 1, 2274: 1, ...

66000 rows × 7 columns

Este bloque convierte los vectores de características de formato disperso a un formato denso, adecuado para el entrenamiento del modelo. La función make_dense toma un diccionario de características (feats) y crea un array de ceros del tamaño del vocabulario, asignando las frecuencias de los tokens en las posiciones correspondientes. Luego, se aplica esta función a cada fila del DataFrame para generar una matriz X_train que contiene los vectores de características densos. También se extraen las etiquetas de clase (y_train) y se ajustan los índices para que comiencen desde 0. Finalmente, X_train y y_train se convierten a tensores de PyTorch, preparándolos para el entrenamiento del modelo en la siguiente etapa.

In [11]:
```
def make_dense(feats):
    x = np.zeros(vocabulary_size)
    for k,v in feats.items():
        x[k] = v
    return x

X_train = np.stack(train_df['features'].progress_map(make_dense))
y_train = train_df['class index'].to_numpy() - 1

X_train = torch.tensor(X_train, dtype=torch.float32)
y_train = torch.tensor(y_train)
```

```
  0%|          | 0/66000 [00:00<?, ?it/s]
```

En este bloque se define y entrena el modelo de clasificación con PyTorch. Se configuran los hiperparámetros (tasa de aprendizaje, épocas, etc.) y se inicializa el modelo como una capa lineal que predice probabilidades para cada clase. Se usa la pérdida de entropía cruzada y el optimizador SGD para ajustar los parámetros. Durante el entrenamiento, los datos se mezclan y se procesan en cada época: se calculan predicciones, se mide la pérdida y se

retropropaga el error para actualizar los pesos, repitiendo el proceso para mejorar el rendimiento del modelo.

In [12]:
```
from torch import nn
from torch import optim

# hyperparameters
lr = 1.0
n_epochs = 5
n_examples = X_train.shape[0]
n_feats = X_train.shape[1]
n_classes = len(labels)

# initialize the model, loss function, optimizer, and data-loader
model = nn.Linear(n_feats, n_classes).to(device)
loss_func = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=lr)

# train the model
indices = np.arange(n_examples)
for epoch in range(n_epochs):
    np.random.shuffle(indices)
    for i in tqdm(indices, desc=f'epoch {epoch+1}'):
        # clear gradients
        model.zero_grad()
        # send datum to right device
        x = X_train[i].unsqueeze(0).to(device)
        y_true = y_train[i].unsqueeze(0).to(device)
        # predict label scores
        y_pred = model(x)
        # compute loss
        loss = loss_func(y_pred, y_true)
        # backpropagate
        loss.backward()
        # optimize model parameters
        optimizer.step()
```

```
epoch 1:   0%|          | 0/66000 [00:00<?, ?it/s]
epoch 2:   0%|          | 0/66000 [00:00<?, ?it/s]
epoch 3:   0%|          | 0/66000 [00:00<?, ?it/s]
epoch 4:   0%|          | 0/66000 [00:00<?, ?it/s]
epoch 5:   0%|          | 0/66000 [00:00<?, ?it/s]
```

Next, we evaluate on the test dataset

Este bloque repite el preprocesamiento del conjunto de entrenamiento para el conjunto de prueba. Se carga el dataset de prueba, se toma una muestra del 50% y se asignan nombres a las columnas. Luego, se limpian y combinan los textos ('title' y 'description'), se tokenizan, y se crean vectores de características usando las mismas funciones definidas antes. Finalmente, los vectores de características (X_test) y las etiquetas (y_test) se convierten a tensores de PyTorch, dejándolos listos para evaluar el rendimiento del modelo.

In [13]:
```
# repeat all preprocessing done above, this time on the test set
test_df = pd.read_csv('test.csv', header=None)

test_df = test_df.sample(frac=0.5, random_state=42)

test_df.columns = ['class index', 'title', 'description']
test_df['text'] = test_df['title'].str.lower() + " " + test_df['description'].str.l
test_df['text'] = test_df['text'].str.replace('\\', ' ', regex=False)
test_df['tokens'] = test_df['text'].progress_map(word_tokenize)
test_df['features'] = test_df['tokens'].progress_map(make_feature_vector)

X_test = np.stack(test_df['features'].progress_map(make_dense))
y_test = test_df['class index'].to_numpy() - 1
X_test = torch.tensor(X_test, dtype=torch.float32)
y_test = torch.tensor(y_test)
```

```
  0%|          | 0/3800 [00:00<?, ?it/s]
  0%|          | 0/3800 [00:00<?, ?it/s]
  0%|          | 0/3800 [00:00<?, ?it/s]
```

Este bloque evalúa el rendimiento del modelo en el conjunto de prueba. Primero, se cambia el modelo a modo de evaluación (model.eval()) para desactivar ciertas operaciones usadas solo durante el entrenamiento. Luego, se desactivan los cálculos de gradientes con torch.no_grad() para ahorrar memoria y acelerar la inferencia. Se envían los datos de prueba a la GPU (si está disponible) y se generan predicciones con el modelo. Finalmente, se usa classification_report de sklearn para imprimir métricas como precisión, recall y F1-score, proporcionando una evaluación detallada del rendimiento del modelo por clase.

In [14]:
```
from sklearn.metrics import classification_report

# set model to evaluation mode
model.eval()

# don't store gradients
with torch.no_grad():
    X_test = X_test.to(device)
    y_pred = torch.argmax(model(X_test), dim=1)
    y_pred = y_pred.cpu().numpy()
    print(classification_report(y_test, y_pred, target_names=labels))
```

```
              precision    recall  f1-score   support

       World       0.92      0.88      0.90       945
      Sports       0.97      0.93      0.95       962
    Business       0.83      0.84      0.84       909
    Sci/Tech       0.84      0.90      0.87       984

    accuracy                           0.89      3800
   macro avg       0.89      0.89      0.89      3800
weighted avg       0.89      0.89      0.89      3800
```