Multiclass Text Classification with¶

Feed-forward Neural Networks and Word Embeddings¶

First, we will do some initialization.

In [9]:

```
import random
import torch
import numpy as np
import pandas as pd
from tqdm.notebook import tqdm

# Habilita tqdm en pandas
tqdm.pandas()

# Pones en True para poder usar la gpu (Si hay una disponible)
use_gpu = True

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

# Semilla random
seed = 1234

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

device: cpu
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.

In [10]:

```
train_df =
pd.read_csv('/kaggle/input/agnews-pytorch-simple-embed-classif-90/AG_NEWS/train.csv',
header=None) # leer el dataset que se usara
train_df.columns = ['class index', 'title', 'description'] # Crear las columnas que se usaran
train_df = train_df.sample(frac = 0.7, random_state = 42) # Elejir una fraccion de los datos
train_df
```

Out[10]:

	class index	title	description
71787	3	BBC set for major shake-up, claims newspaper	London - The British Broadcasting Corporation,...

	class index	title	description
67218	3	Marsh averts cash crunch	Embattled insurance broker #39;s banks agree t...
54066	2	Jeter, Yankees Look to Take Control (AP)	AP - Derek Jeter turned a season that started ...
7168	4	Flying the Sun to Safety	When the Genesis capsule comes back to Earth w...
29618	3	Stocks Seen Flat as Nortel and Oil Weigh	NEW YORK (Reuters) - U.S. stocks were set to ...
...	...	...	...
53857	1	FDA Accused of Silencing Vioxx Warnings	WASHINGTON - The Food and Drug Administration ...
111476	2	Buckeyes won #39;t play in NCAA or NIT tourneys	COLUMBUS, Ohio Ohio State has sanctioned its m...
6343	3	Rate hikes by Fed work in two ways	If you #39;ve noticed that the price of everyt...
20736	4	NASA Administrator Offers Support for Kennedy ...	The following is a statement from NASA Adminis...

	class index	title	description
34378	2	Twins make it 3 straight	The Minnesota Twins clinched on a bus in 1991....

84000 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.

In [11]:

```
labels = open('/kaggle/input/classes/classes.txt').read().splitlines() # Crear lables para almacenar todos los nombres de las clases
classes = train_df['class index'].map(lambda i: labels[i-1]) # Crear clases para almacenar todos los nombres de las clases
train_df.insert(1, 'class', classes) # Insertar los nombres de las clases en el data frame
train_df
```

Out[11]:

	class index	class	title	description
71787	3	Business	BBC set for major shake-up, claims	London - The British Broadcasting

	class index	class	title	description
			newspaper	Corporation,...
67218	3	Business	Marsh averts cash crunch	Embattled insurance broker #39;s banks agree t...
54066	2	Sports	Jeter, Yankees Look to Take Control (AP)	AP - Derek Jeter turned a season that started ...
7168	4	Sci/Tech	Flying the Sun to Safety	When the Genesis capsule comes back to Earth w...
29618	3	Business	Stocks Seen Flat as Nortel and Oil Weigh	NEW YORK (Reuters) - U.S. stocks were set to ...
...	...	...	...	...
53857	1	World	FDA Accused of Silencing Vioxx Warnings	WASHINGTON - The Food and Drug Administration ...
111476	2	Sports	Buckeyes won #39;t play in NCAA or NIT tourneys	COLUMBUS, Ohio Ohio State has sanctioned its m...

	class index	class	title	description
6343	3	Business	Rate hikes by Fed work in two ways	If you #39;ve noticed that the price of everyt...
20736	4	Sci/Tech	NASA Administrator Offers Support for Kennedy ...	The following is a statement from NASA Adminis...
34378	2	Sports	Twins make it 3 straight	The Minnesota Twins clinched on a bus in 1991....

84000 rows × 4 columns

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

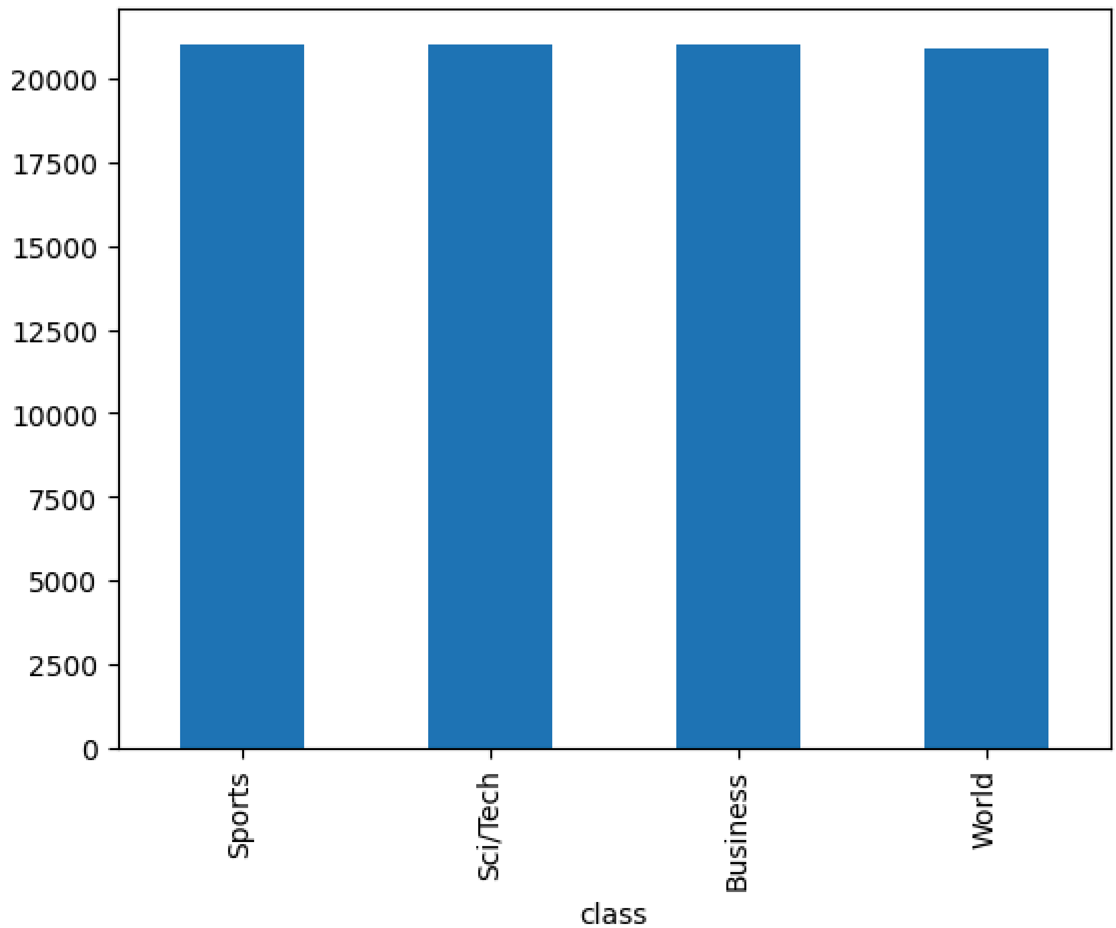
In [12]:

```
pd.value_counts(train_df['class']).plot.bar() # Se grafica pra ver como estan los resultados
```

```
/tmp/ipykernel_30/1245903889.py:1: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  pd.value_counts(train_df['class']).plot.bar()
```

Out[12]:

<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".

In [13]:

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

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

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

In [14]:

```
train_df['text'] = train_df['title'].str.lower() + " " + train_df['description'].str.lower() # Combina las columnas description y titulo, las juntas en un nueva cokumna llamada text y las pasas a minusculas
train_df['text'] = train_df['text'].str.replace('\\', ' ', regex=False) # Quita los \ y los pone por espacios en blanco
train_df
```

Out[14]:

	class index	class	title	description	text
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...

	class index	class	title	description	text
67218	3	Business	Marsh averts cash crunch	Embattled insurance broker #39;s banks agree t...	marsh averts cash crunch embattled insurance b...
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 - ...
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...
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 ...
...	...	...	...	...	...
53857	1	World	FDA Accused of Silencing Vioxx Warnings	WASHINGTON - The Food and Drug Administration ...	fda accused of silencing vioxx warnings washin...
111476	2	Sports	Buckeyes won #39;t	COLUMBUS, Ohio	buckeyes won #39;t

	class index	class	title	description	text
			play in NCAA or NIT tourneys	Ohio State has sanctioned its m...	play in ncaa or nit tourney...
6343	3	Business	Rate hikes by Fed work in two ways	If you #39;ve noticed that the price of everyt...	rate hikes by fed work in two ways if you #39;...
20736	4	Sci/Tech	NASA Administrator Offers Support for Kennedy ...	The following is a statement from NASA Adminis...	nasa administrator offers support for kennedy ...
34378	2	Sports	Twins make it 3 straight	The Minnesota Twins clinched on a bus in 1991....	twins make it 3 straight the minnesota twins c...

84000 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.

In [15]:

```
from nltk.tokenize import word_tokenize

train_df['tokens'] = train_df['text'].progress_map(word_tokenize) # Tokenizacion de palabras y los almacena en una nueva columna
train_df
```

0%| | 0/84000 [00:00<?, ?it/s]

Out[15]:

	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	When the	flying the	[flying,

	class index	class	title	description	text	tokens
			Sun to Safety	Genesis capsule comes back to Earth w...	sun to safety when the genesis caps...	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...
...	...	...	...	...	...	...
53857	1	World	FDA Accused of Silencing Vioxx Warnings	WASHINGTON - The Food and Drug Administration ...	fda accused of silencing vioxx warnings washin...	[fda, accused, of, silencing, vioxx, warnings, ...
111476	2	Sports	Buckeyes won #39;t play in NCAA or NIT tourneys	COLUMBUS, Ohio Ohio State has sanctioned its m...	buckeyes won #39;t play in ncaa or nit tourney...	[buckeyes, won, #, 39, ;, t, play, in, ncaa, o...
6343	3	Business	Rate hikes by Fed work in two ways	If you #39;ve noticed that the price of everyt...	rate hikes by fed work in two ways if you #39;...	[rate, hikes, by, fed, work, in, two, ways, if...

	class index	class	title	description	text	tokens
20736	4	Sci/Tech	NASA Administrator Offers Support for Kennedy ...	The following is a statement from NASA Adminis...	nasa administrator offers support for kennedy ...	[nasa, administrator, offers, support, for, ke...
34378	2	Sports	Twins make it 3 straight	The Minnesota Twins clinched on a bus in 1991....	twins make it 3 straight the minnesota twins c...	[twins, make, it, 3, straight, the, minnesota,...

84000 rows × 6 columns

Now we will load the GloVe word embeddings.

In [16]:

```
from gensim.models import KeyedVectors
glove = KeyedVectors.load_word2vec_format("/kaggle/input/glove-fasttext-embedding-for-medium-articles/glove.6B.300d.txt", no_header=True) # Importamos el dataset de glove
glove.vectors.shape
```

Out[16]:

(400000, 300)

The word embeddings have been pretrained in a different corpus, so it would be a good idea to estimate how good our tokenization matches the GloVe vocabulary.

In [17]:

```
from collections import Counter

# Funcion que cuenta las palabras desconocidas (no incluidas en el vocabulario) en el dataset
def count_unknown_words(data, vocabulary):
    counter = Counter()
    for row in tqdm(data):
        counter.update(tok for tok in row if tok not in vocabulary)
    return counter

# Encuentra la cantidad de veces que cada token desconocido aparece en el corpus
c = count_unknown_words(train_df['tokens'], glove.key_to_index)

# Encuentra el número total de tokens en el corpus
total_tokens = train_df['tokens'].map(len).sum()

# Calcula estadísticas sobre la aparición de tokens desconocidos
unk_tokens = sum(c.values())
percent_unk = unk_tokens / total_tokens
distinct_tokens = len(list(c))

# Imprime las estadísticas del corpus
print(f'total number of tokens: {total_tokens:,}')
print(f'number of unknown tokens: {unk_tokens:,}')
print(f'number of distinct unknown tokens: {distinct_tokens:,}')
print(f'percentage of unkown tokens: {percent_unk:.2%}')
print('top 50 unknown words:')
for token, n in c.most_common(10):
    print(f'\t{n}\t{token}')
```

```
  0%|          | 0/84000 [00:00<?, ?it/s]
```

```
total number of tokens: 3,691,911
number of unknown tokens: 46,427
number of distinct unknown tokens: 18,956
percentage of unkown tokens: 1.26%
top 50 unknown words:
	2055	/b
	1502	href=
	1501	/a
	1280	//www.investor.reuters.com/fullquote.aspx
	1280	target=/stocks/quickinfo/fullquote
	417	/p
	356	newsfactor
	340	cbs.mw
	300	color=
	291	face=
```

Glove embeddings seem to have a good coverage on this dataset -- only 1.25% of the tokens in the dataset are unknown, i.e., don't appear in the GloVe vocabulary.

Still, we will need a way to handle these unknown tokens. Our approach will be to add a new embedding to GloVe that will be used to represent them. This new embedding will be initialized as the average of all the GloVe embeddings.

We will also add another embedding, this one initialized to zeros, that will be used to pad the sequences of tokens so that they all have the same length. This will be useful when we train with mini-batches.

In [18]:

```
# Valores de cadena que corresponden a los nuevos embeddings
unk_tok = '[UNK]'
pad_tok = '[PAD]'
```

```python
# Inicializa los valores para los nuevos embeddings
unk_emb = glove.vectors.mean(axis=0)
pad_emb = np.zeros(300)

# Agrega los nuevos embeddings al modelo glove
glove.add_vectors([unk_tok, pad_tok], [unk_emb, pad_emb])

# Obtiene los IDs de los tokens correspondientes a los nuevos embeddings
unk_id = glove.key_to_index[unk_tok]
pad_id = glove.key_to_index[pad_tok]

unk_id, pad_id
```

Out[18]:

(400000, 400001)

In [19]:

```python
from sklearn.model_selection import train_test_split

train_df, dev_df = train_test_split(train_df, train_size=0.8) # Elejir una fraccion de los datos
train_df.reset_index(inplace=True) # Reinicia los índices en los dataframes para que comiencen desde 0 después de la división

dev_df.reset_index(inplace=True)
```

We will now add a new column to our dataframe that will contain the padded sequences of token ids.

In [20]:

```
threshold = 10
tokens = train_df['tokens'].explode().value_counts() # Cuenta la frecuencia de cada token en la columna de tokens
vocabulary = set(tokens[tokens > threshold].index.tolist()) # Crea el vocabulario con los tokens que supere la frecuencaia
print(f'vocabulary size: {len(vocabulary):,}') # Imprime los resultados
```

vocabulary size: 14,309

In [21]:

```
# Encuentra la logitud más larga de la lista de tokens
max_tokens = train_df['tokens'].map(len).max()

# Retorna unk_id para los tokens infrecuentes
def get_id(tok):
    if tok in vocabulary:
        return glove.key_to_index.get(tok, unk_id)
    else:
        return unk_id

# Función que recibe una lista de tokens y devuelve una lista de ids de tokens,
# agregando padding según la longitud máxima establecida
def token_ids(tokens):
    tok_ids = [get_id(tok) for tok in tokens]
    pad_len = max_tokens - len(tok_ids)
    return tok_ids + [pad_id] * pad_len

# Añade la nueva columna al data frame
train_df['token ids'] = train_df['tokens'].progress_map(token_ids)
train_df
```

0%| | 0/67200 [00:00<?, ?it/s]

Out[21]:

	index	class index	class	title	description	text	tokens	token ids
0	109275	4	Sci/Tech	Mmo2, Lucent to deploy converged fixed-mobile ...	UK mobile operator Mmo2 and US telecoms equipm...	mmo2, lucent to deploy converged fixed-mobile ...	[mmo2, ,, lucent, to, deploy, converged, fixed...	[122597, 1, 15725, 4, 8169, 21252, 400000, 849...
1	89047	3	Business	Spitzer Plans to Sue Insurer	New York Attorney General Eliot Spitzer will f...	spitzer plans to sue insurer new york attorney...	[spitzer, plans, to, sue, insurer, new, york, ...	[12185, 559, 4, 6415, 10646, 50, 196, 1223, 21...
2	118050	1	World	Britain Cannot Detain Terror Suspects Indefini...	Nine Law Lords ruled in favour of a group of m...	britain cannot detain terror suspects indefini...	[britain, can, not, detain, terror, suspects, ...	[695, 86, 36, 14097, 1974, 2330, 9595, 45, 202...
3	106813	1	World	Belgrade attack #39;was	A feared assassination	belgrade attack #39;was	[belgrade, attack, #, 39, ;, was,	[4038, 436, 2749, 3403, 89, 15,

	index	class index	class	title	description	text	tokens	token ids
				road rage #39;	attempt on Serbia #39;s...	road rage #39; a fear...	road, rage, ...	586, 9012, 274...
4	84844	3	Business	Arctic Thaw Threatens People, Polar Bears	OSLO (Reuters) - Global warming is heating th...	arctic thaw threatens people, polar bears osl...	[arctic, thaw, threatens, people, ,, polar, be...	[7574, 20189, 6805, 69, 1, 10158, 4509, 6737, ...
...	...	...	...	...	...	...	...	...
67195	67493	3	Business	Jeans Maker VF Sees Earns Up 24 Percent (Reuters)	Reuters - VF Corp , the world's largest \jeans ...	jeans maker vf sees earns up 24 percent (reute...	[jeans, maker, vf, sees, earns, up, 24, percen...	[400000, 2737, 400000, 3109, 12803, 60, 795, 7...
67196	58333	3	Business	Temasek Makes S\$7.4 Bln Profit, Gets Top AAA ...	Temasek Holdings Pte earned S\$7.4 billion (\$...	temasek makes $7.4 bln profit, gets top aaa ...	[temasek, makes, s, $, 7.4, bln, profit, ,, ge...	[400000, 907, 1534, 80, 14321, 17494, 1269, 1,...

	index	class index	class	title	description	text	tokens	token ids
67197	112554	3	Business	Local gamer : Grand Theft Auto #39; steals the ...	Just how excited is Justin Field about the new...	local gamer : grand theft auto #39; steals the ...	[local, gamer, :, grand, theft, auto, #, 39, ;...	[250, 400000, 45, 1063, 6539, 2612, 2749, 3403...
67198	116840	3	Business	Sprint, Nextel Agree To Merge	The deal, valued at \$35 billion, will create ...	sprint, nextel agree to merge the deal, valued ...	[sprint, ,, nextel, agree, to, merge, , the, dea...	[5514, 1, 17774, 2137, 4, 9194, 0, 435, 1, 595...
67199	34067	3	Business	Export Cut to China Seen as Clever Strategy on...	Yukos, the Russian oil giant, is playing a wea...	export cut to china seen as clever strategy on...	[export, cut, to, china, seen, as, clever, str...	[2467, 611, 4, 132, 541, 19, 11114, 1747, 13, ...

67200 rows × 8 columns

In [22]:

```
max_tokens = dev_df['tokens'].map(len).max() # Encuentra la longitud de la lista de tokens
```

más larga
dev_df['token ids'] **=** dev_df['tokens']**.**progress_map(token_ids) # *Agrega una nueva columna al data frame con los ids de tokens*

dev_df

0%| | 0/16800 [00:00<?, ?it/s]

Out[22]:

	index	class index	class	title	description	text	tokens	token ids
0	111352	4	Sci/Tech	Canon loses printer recycling case	Refilling, reselling cartridges doesn't violat...	canon loses printer recycling case refilling, ...	[canon, loses, printer, recycling, case, refil...	[9579, 7233, 13568, 12520, 305, 400000, 1, 400...
1	102053	4	Sci/Tech	'EICU' Lets Doctors Monitor Many Patients (AP)	AP - Your next doctor could be keeping an eye ...	'eicu' lets doctors monitor many patients (ap)...	['eicu, ', lets, doctors, monitor, many, patie...	[400000, 57, 8235, 1768, 3933, 109, 1615, 23, ...
2	50868	4	Sci/Tech	Yahoo CEO Sees No Need	Reuters - In an era of wides	yahoo ceo sees no need	[yahoo, ceo, sees, no, need,	[6600, 3695, 3109, 84, 408, 4,

	index	class index	class	title	description	text	tokens	token ids
				to Join Media Merger Fr...	pread media\ consol ...	to join media merger fr...	to, join, media, ...	1429, 493, 3176...
3	27469	2	Sports	Sports view: Chargers Are Surprise Winners (AP)	AP - So the San Diego Chargers shocked the NFL...	sports view: chargers are surprise winners (ap)...	[sports view, :, chargers, are, surprise, winne...	[400000, 45, 12104, 32, 2661, 2945, 23, 1582, ...
4	66091	3	Business	Stocks Fall on J.P. Morgan Chase and Oil	NEW YORK (Reuters) - U.S. stocks fell on Wedn...	stocks fall on j.p. morgan chase and oil new ...	[stocks, fall, on, j.p., morgan, chase, and, o...	[895, 807, 13, 12227, 3123, 4212, 5, 316, 50, ...
...	...	...	...	...	...	...	...	...
16795	109691	1	World	Former Marine Testifies to Atrocities in Iraq	A former U.S. Marine staff sergeant testified ...	former marine testifies to atrocities in iraq ...	[former, marine, testifies, to, atrocities, in...	[157, 2266, 27149, 4, 8088, 6, 233, 7, 157, 99...

	index	class index	class	title	description	text	tokens	token ids
16796	35541	4	Sci/Tech	Blogging the Story Alive	Bloggers force CBS News to admit to a serious ...	blogging the story alive bloggers force cbs ne...	[blogging, the, story, alive, bloggers, force,...	[30031, 0, 523, 2977, 19305, 352, 3286, 172, 4...
16797	106135	3	Business	Getting your report	Consumers in Arizona and 12 other Western stat...	getting your report consumers in arizona and 1...	[getting, your, report, consumers, in, arizona...	[881, 392, 255, 2034, 6, 2203, 5, 421, 68, 556...
16798	61875	3	Business	GM reports poor quarterly profits	DETROIT: General Motors Corp posted on Thursda...	gm reports poor quarterly profits detroit: gen...	[gm, reports, poor, quarterly, profits, detroi...	[2907, 687, 992, 6206, 2243, 2369, 45, 216, 46...
16799	40321	3	Business	For Cingular, Becoming No. 1	The union of Cingular and AT T	for cingular, becoming no. 1 also	[for, cingular, ,, becoming, no, .,	[10, 31779, 1, 1663, 84, 2, 176,

index	class index	class	title	description	text	tokens	token ids
			Also Poses Risks	Wireless would ...	poses risks ...	1, also, p...	52, 9734, 334...

16800 rows × 8 columns

Now we will get a numpy 2-dimensional array corresponding to the token ids, and a 1-dimensional array with the gold classes. Note that the classes are one-based (i.e., they start at one), but we need them to be zero-based, so we need to subtract one from this array.

In [23]:

```
from torch.utils.data import Dataset

# Creas las clase de MyDataset
class MyDataset(Dataset):
    def __init__(self, x, y):
        self.x = x
        self.y = y

    # Devuelve la longitud del dataset
    def __len__(self):
        return len(self.y)

    # Obtiene el elemento en la posición index y lo convierte en un tensor de PyTorch
    def __getitem__(self, index):
        x = torch.tensor(self.x[index])
        y = torch.tensor(self.y[index])
        return x, y
```

Next, we construct our PyTorch model, which is a feed-forward neural network with two layers:

In [24]:

```
from torch import nn
import torch.nn.functional as F

# Creas la clase Model
class Model(nn.Module):
    def __init__(self, vectors, pad_id, hidden_dim, output_dim, dropout):
        super().__init__()
        # Verifica si 'vectors' es un tensor, de lo contrario, lo convierte
        if not torch.is_tensor(vectors):
            vectors = torch.tensor(vectors)
        # Almacena el ID del padding
        self.padding_idx = pad_id
        # Crea la capa de embeddings a partir de los vectores preentrenados
        self.embs = nn.Embedding.from_pretrained(vectors, padding_idx=pad_id)
        # Define las capas feedforward en una secuencia
        self.layers = nn.Sequential(
            nn.Dropout(dropout),
            nn.Linear(vectors.shape[1], hidden_dim),
            nn.ReLU(),
            nn.Dropout(dropout),
            nn.Linear(hidden_dim, output_dim),
        )

    def forward(self, x):
        # Obtiene un arreglo booleano donde los elementos de padding son marcados como
False
        not_padding = torch.isin(x, self.padding_idx, invert=True)
        # Calcula las longitudes de los ejemplos (excluyendo el padding)
        lengths = torch.count_nonzero(not_padding, axis=1)
        # Obtiene los embeddings para la entrada
        x = self.embs(x)
        # Calcula la media de los embeddings
        x = x.sum(dim=1) / lengths.unsqueeze(dim=1)
        # Pasa el resultado al resto del modelo
        output = self.layers(x)
```

```
        # Calcula softmax si no estamos en modo de entrenamiento
        #if not self.training:
        #    output = F.softmax(output, dim=1)
        return output
```

Next, we implement the training procedure. We compute the loss and accuracy on the development partition after each epoch.

In [25]:

```
from torch import optim
from torch.utils.data import DataLoader
from sklearn.metrics import accuracy_score

# Hiperparámetros
lr = 1e-3
weight_decay = 0
batch_size = 500
shuffle = True
n_epochs = 5
hidden_dim = 50
output_dim = len(labels)
dropout = 0.1
vectors = glove.vectors

# Inicializa el modelo, la función de pérdida, el optimizador y el cargador de datos
model = Model(vectors, pad_id, hidden_dim, output_dim, dropout).to(device)
loss_func = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=lr, weight_decay=weight_decay)
train_ds = MyDataset(train_df['token ids'], train_df['class index'] - 1)
train_dl = DataLoader(train_ds, batch_size=batch_size, shuffle=shuffle)
dev_ds = MyDataset(dev_df['token ids'], dev_df['class index'] - 1)
dev_dl = DataLoader(dev_ds, batch_size=batch_size, shuffle=shuffle)

# Listas para almacenar pérdidas y precisiones de entrenamiento y desarrollo
train_loss = []
train_acc = []
```

```python
dev_loss = []
dev_acc = []

# Entrena el modelo
for epoch in range(n_epochs):
    losses = []
    gold = []
    pred = []
    model.train()
    for X, y_true in tqdm(train_dl, desc=f'epoch {epoch+1} (train)'):
        # Limpia los gradientes
        model.zero_grad()
        # Envía el lote al dispositivo correcto
        X = X.to(device)
        y_true = y_true.to(device)
        # Predice las puntuaciones de las etiquetas
        y_pred = model(X)
        # Calcula la pérdida
        loss = loss_func(y_pred, y_true)
        # Acumula para graficar
        losses.append(loss.detach().cpu().item())
        gold.append(y_true.detach().cpu().numpy())
        pred.append(np.argmax(y_pred.detach().cpu().numpy(), axis=1))
        # Realiza el backpropagate
        loss.backward()
        #Optimiza los parámetros del modelo
        optimizer.step()
    train_loss.append(np.mean(losses))
    train_acc.append(accuracy_score(np.concatenate(gold), np.concatenate(pred)))

    model.eval()  # Establece el modelo en modo de evaluación
    with torch.no_grad():
        losses = []
        gold = []
        pred = []
        for X, y_true in tqdm(dev_dl, desc=f'epoch {epoch+1} (dev)'):
            X = X.to(device)
            y_true = y_true.to(device)
            y_pred = model(X)
            loss = loss_func(y_pred, y_true)
            losses.append(loss.cpu().item())
            gold.append(y_true.cpu().numpy())
            pred.append(np.argmax(y_pred.cpu().numpy(), axis=1))
        # Almacena la pérdida y precisión promedio para el conjunto de desarrollo
        dev_loss.append(np.mean(losses))
        dev_acc.append(accuracy_score(np.concatenate(gold), np.concatenate(pred)))
```

```
epoch 1 (train):   0%|          | 0/135 [00:00<?, ?it/s]

epoch 1 (dev):   0%|          | 0/34 [00:00<?, ?it/s]

epoch 2 (train):   0%|          | 0/135 [00:00<?, ?it/s]

epoch 2 (dev):   0%|          | 0/34 [00:00<?, ?it/s]

epoch 3 (train):   0%|          | 0/135 [00:00<?, ?it/s]

epoch 3 (dev):   0%|          | 0/34 [00:00<?, ?it/s]

epoch 4 (train):   0%|          | 0/135 [00:00<?, ?it/s]

epoch 4 (dev):   0%|          | 0/34 [00:00<?, ?it/s]

epoch 5 (train):   0%|          | 0/135 [00:00<?, ?it/s]

epoch 5 (dev):   0%|          | 0/34 [00:00<?, ?it/s]
```

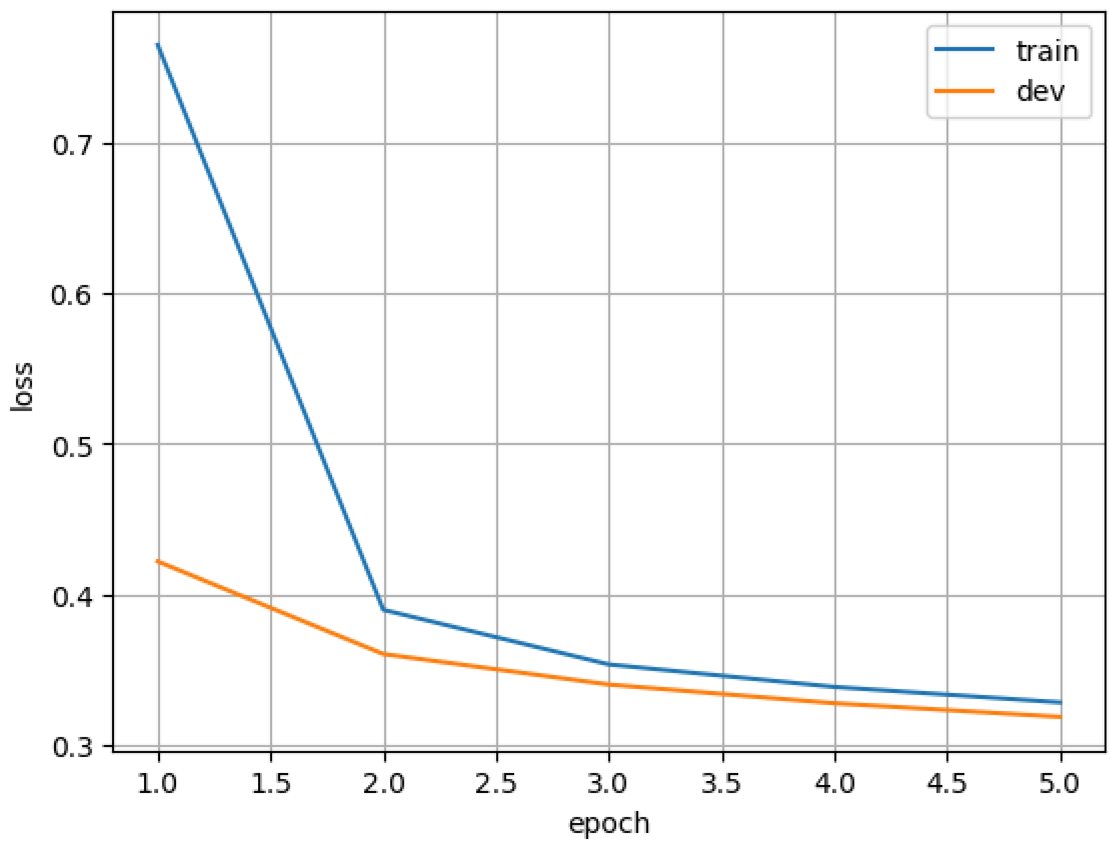
Let's plot the loss and accuracy on dev:

In [26]:

```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
# Se crean graficas para ver los resultados de train_loss y dev_loss
x = np.arange(n_epochs) + 1

plt.plot(x, train_loss)
plt.plot(x, dev_loss)
plt.legend(['train', 'dev'])
plt.xlabel('epoch')
plt.ylabel('loss')
plt.grid(True)
```

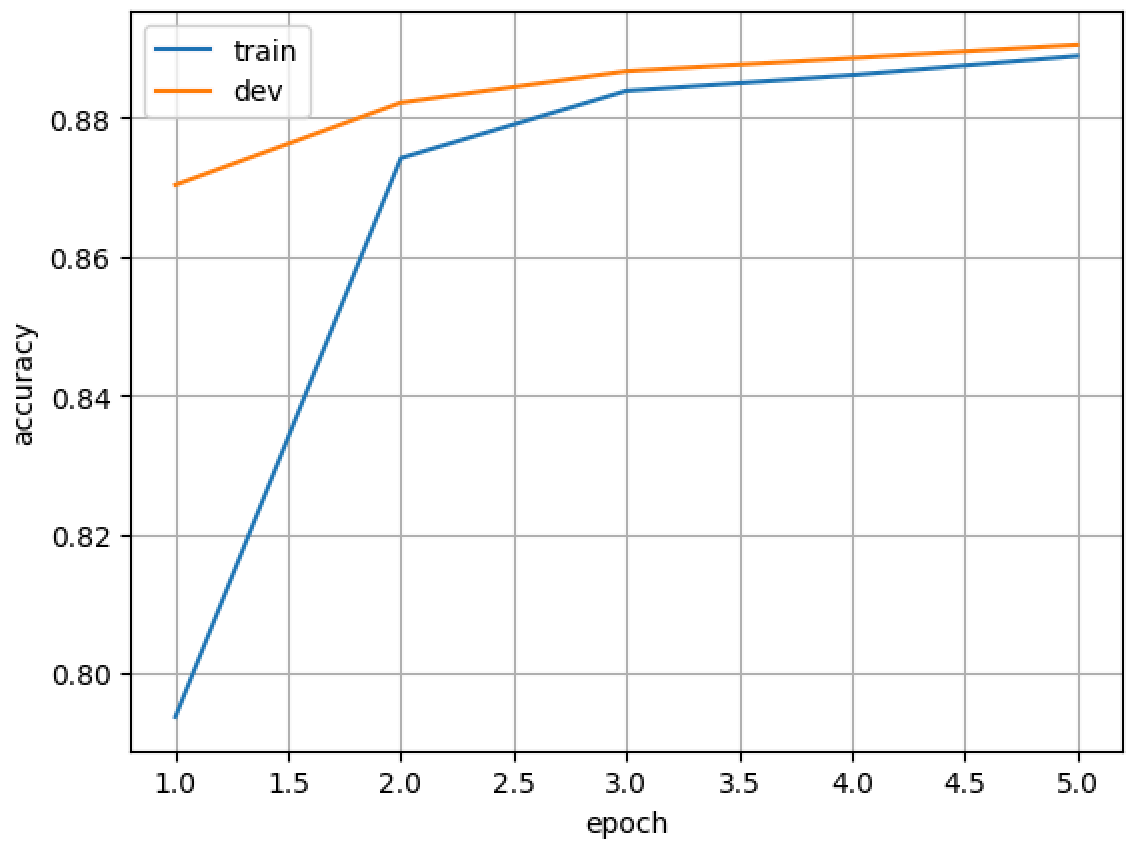


In [27]:

```
# Se crean graficas para ver los resultados de train_acc y dev_acc

plt.plot(x, train_acc)
```

```
plt.plot(x, dev_acc)
plt.legend(['train', 'dev'])
plt.xlabel('epoch')
plt.ylabel('accuracy')
plt.grid(True)
```



Next, we evaluate on the testing partition:

In [28]:

```
# Repite todo el preproceso de arriba, pero ahora con el data set de test
```

```
test_df =
pd.read_csv('/kaggle/input/agnews-pytorch-simple-embed-classif-90/AG_NEWS/test.csv',
header=None)
test_df.columns = ['class index', 'title', 'description']
test_df['text'] = test_df['title'].str.lower() + " " + test_df['description'].str.lower()
test_df['text'] = test_df['text'].str.replace('\\', ' ', regex=False)
test_df['tokens'] = test_df['text'].progress_map(word_tokenize)
max_tokens = dev_df['tokens'].map(len).max()
test_df['token ids'] = test_df['tokens'].progress_map(token_ids)
```

```
0%|          | 0/7600 [00:00<?, ?it/s]
```

```
0%|          | 0/7600 [00:00<?, ?it/s]
```

In [29]:

```
from sklearn.metrics import classification_report

# Se pone el modelo en modo evaluacion
model.eval()

dataset = MyDataset(test_df['token ids'], test_df['class index'] - 1)
data_loader = DataLoader(dataset, batch_size=batch_size)
y_pred = []

# No se guardan los gradientes
with torch.no_grad():
    for X, _ in tqdm(data_loader):  # Itera sobre los lotes en el DataLoader
        X = X.to(device) # Envía los datos al dispositivo (CPU o GPU)
        # Predice la clase más probable para cada ejemplo en el lote
        y = torch.argmax(model(X), dim=1)
        # Convierte el tensor en un array numpy (y lo envía de regreso a la CPU si es necesario)
        y_pred.append(y.cpu().numpy())
        # Imprime los resultados
    print(classification_report(dataset.y, np.concatenate(y_pred), target_names=labels))
```

0%| | 0/16 [00:00<?, ?it/s]

	precision	recall	f1-score	support
World	0.92	0.87	0.90	1900
Sports	0.95	0.97	0.96	1900
Business	0.83	0.86	0.85	1900
Sci/Tech	0.87	0.86	0.86	1900
accuracy			0.89	7600
macro avg	0.89	0.89	0.89	7600
weighted avg	0.89	0.89	0.89	7600