Text Classification Using Transformer Networks (BERT)

Some initialization:

```python
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 '
         print(f'device: {device.type}')

         # random seed
         seed = 1222

         # 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: 1222
```

Read the train/dev/test datasets and create a HuggingFace `Dataset` object:

```python
In [2]:  def read_data(filename):
             # read csv file
             df = pd.read_csv(filename, header=None)
             # add column names
             df.columns = ['label', 'title', 'description']
             # make labels zero-based
             df['label'] -= 1
             # concatenate title and description, and remove backslashes
             df['text'] = df['title'] + " " + df['description']
             df['text'] = df['text'].str.replace('\\', ' ', regex=False)
             return df
```

```python
In [3]:  labels = open('/kaggle/input/namesste/classes.txt').read().splitlines()
         train_df = read_data('/kaggle/input/agnews-pytorch-simple-embed-classif-90/A
```

```
test_df = read_data('/kaggle/input/agnews-pytorch-simple-embed-classif-90/AC
train_df
```

Out[3]:

	label	title	description	text
0	2	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	2	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...	Carlyle Looks Toward Commercial Aerospace (Reu...
2	2	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...	Oil and Economy Cloud Stocks' Outlook (Reuters...
3	2	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...	Iraq Halts Oil Exports from Main Southern Pipe...
4	2	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	0	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	1	Renteria signing a top-shelf deal	Red Sox general manager Theo Epstein acknowled...	Renteria signing a top-shelf deal Red Sox gene...
119997	1	Saban not going to Dolphins yet	The Miami Dolphins will put their courtship of...	Saban not going to Dolphins yet The Miami Dolp...
119998	1	Today's NFL games	PITTSBURGH at NY GIANTS Time: 1:30 p.m. Line: ...	Today's NFL games PITTSBURGH at NY GIANTS Time...
119999	1	Nets get Carter from Raptors	INDIANAPOLIS -- All-Star Vince Carter was trad...	Nets get Carter from Raptors INDIANAPOLIS -- A...

120000 rows × 4 columns

In [4]:
```python
from sklearn.model_selection import train_test_split

train_df, eval_df = train_test_split(train_df, train_size=0.9)
train_df.reset_index(inplace=True, drop=True)
eval_df.reset_index(inplace=True, drop=True)

print(f'train rows: {len(train_df.index):,}')
print(f'eval rows: {len(eval_df.index):,}')
print(f'test rows: {len(test_df.index):,}')
```

```
train rows: 108,000
eval rows: 12,000
test rows: 7,600
```

In [5]:
```python
from datasets import Dataset, DatasetDict

ds = DatasetDict()
ds['train'] = Dataset.from_pandas(train_df)
ds['validation'] = Dataset.from_pandas(eval_df)
ds['test'] = Dataset.from_pandas(test_df)
ds
```

Out[5]:
```
DatasetDict({
    train: Dataset({
        features: ['label', 'title', 'description', 'text'],
        num_rows: 108000
    })
    validation: Dataset({
        features: ['label', 'title', 'description', 'text'],
        num_rows: 12000
    })
    test: Dataset({
        features: ['label', 'title', 'description', 'text'],
        num_rows: 7600
    })
})
```

Tokenize the texts:

In [6]:
```python
from transformers import AutoTokenizer

transformer_name = 'bert-base-cased'
tokenizer = AutoTokenizer.from_pretrained(transformer_name)
```

```
tokenizer_config.json:    0%|          | 0.00/49.0 [00:00<?, ?B/s]
config.json:    0%|          | 0.00/570 [00:00<?, ?B/s]
vocab.txt:    0%|          | 0.00/213k [00:00<?, ?B/s]
tokenizer.json:    0%|          | 0.00/436k [00:00<?, ?B/s]
```

```
/opt/conda/lib/python3.10/site-packages/transformers/tokenization_utils_bas
e.py:1617: FutureWarning: `clean_up_tokenization_spaces` was not set. It wil
l be set to `True` by default. This behavior will be deprecated in transform
ers v4.45, and will be then set to `False` by default. For more details chec
k this issue: https://github.com/huggingface/transformers/issues/31884
  warnings.warn(
```

In [7]:
```python
def tokenize(examples):
    return tokenizer(examples['text'], truncation=True)

train_ds = ds['train'].map(
    tokenize, batched=True,
    remove_columns=['title', 'description', 'text'],
)
eval_ds = ds['validation'].map(
    tokenize,
    batched=True,
    remove_columns=['title', 'description', 'text'],
```

```
)
train_ds.to_pandas()
```

```
Map:     0%|              | 0/108000 [00:00<?, ? examples/s]
Map:     0%|              | 0/12000 [00:00<?, ? examples/s]
```

Out[7]:

	label	input_ids	token_type_ids	attention_mask
0	1	[101, 3458, 108, 3614, 132, 7127, 108, 3614, 1...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
1	0	[101, 158, 119, 156, 119, 20339, 22038, 1116, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
2	0	[101, 10864, 8093, 1116, 15386, 8736, 1130, 23...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
3	2	[101, 1975, 25401, 1116, 5600, 1106, 4348, 321...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
4	2	[101, 140, 17030, 16016, 2896, 12853, 3052, 14...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
...	...	...	...	...
107995	1	[101, 12366, 112, 11491, 1110, 3264, 143, 2456...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
107996	2	[101, 108, 3614, 132, 16409, 22910, 1200, 108,...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
107997	0	[101, 139, 19268, 1120, 8644, 13530, 1107, 212...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
107998	0	[101, 8612, 14099, 10498, 1111, 1148, 1159, 11...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
107999	3	[101, 21906, 14334, 11290, 1116, 2170, 18959, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...

108000 rows × 4 columns

Create the transformer model:

In [8]:
```python
from torch import nn
from transformers.modeling_outputs import SequenceClassifierOutput
from transformers.models.bert.modeling_bert import BertModel, BertPreTrained

# https://github.com/huggingface/transformers/blob/65659a29cf5a079842e61a63c

class BertForSequenceClassification(BertPreTrainedModel):
```

```python
    def __init__(self, config):
        super().__init__(config)
        self.num_labels = config.num_labels
        self.bert = BertModel(config)
        self.dropout = nn.Dropout(config.hidden_dropout_prob)
        self.classifier = nn.Linear(config.hidden_size, config.num_labels)
        self.init_weights()

    def forward(self, input_ids=None, attention_mask=None, token_type_ids=No
        outputs = self.bert(
            input_ids,
            attention_mask=attention_mask,
            token_type_ids=token_type_ids,
            **kwargs,
        )
        cls_outputs = outputs.last_hidden_state[:, 0, :]
        cls_outputs = self.dropout(cls_outputs)
        logits = self.classifier(cls_outputs)
        loss = None
        if labels is not None:
            loss_fn = nn.CrossEntropyLoss()
            loss = loss_fn(logits, labels)
        return SequenceClassifierOutput(
            loss=loss,
            logits=logits,
            hidden_states=outputs.hidden_states,
            attentions=outputs.attentions,
        )
```

In [9]:
```python
from transformers import AutoConfig

config = AutoConfig.from_pretrained(
    transformer_name,
    num_labels=len(labels),
)

model = (
    BertForSequenceClassification
    .from_pretrained(transformer_name, config=config)
)
```

```
model.safetensors:   0%|              | 0.00/436M [00:00<?, ?B/s]
```

```
Some weights of BertForSequenceClassification were not initialized from the
model checkpoint at bert-base-cased and are newly initialized: ['classifier.
bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use
it for predictions and inference.
```

Create the trainer object and train:

In [10]:
```python
from transformers import TrainingArguments

num_epochs = 2
batch_size = 24
weight_decay = 0.01
```

```python
model_name = f'{transformer_name}-sequence-classification'

training_args = TrainingArguments(
    output_dir=model_name,
    log_level='error',
    num_train_epochs=num_epochs,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    evaluation_strategy='epoch',
    weight_decay=weight_decay,
)
```

```
/opt/conda/lib/python3.10/site-packages/transformers/training_args.py:1545:
FutureWarning: `evaluation_strategy` is deprecated and will be removed in ve
rsion 4.46 of 🤗 Transformers. Use `eval_strategy` instead
  warnings.warn(
```

In [11]:
```python
from sklearn.metrics import accuracy_score

def compute_metrics(eval_pred):
    y_true = eval_pred.label_ids
    y_pred = np.argmax(eval_pred.predictions, axis=-1)
    return {'accuracy': accuracy_score(y_true, y_pred)}
```

In [12]:
```python
from transformers import Trainer

trainer = Trainer(
    model=model,
    args=training_args,
    compute_metrics=compute_metrics,
    train_dataset=train_ds,
    eval_dataset=eval_ds,
    tokenizer=tokenizer,
)
```

In [13]:
```python
trainer.train()
```

```
wandb: WARNING The `run_name` is currently set to the same value as `Trainin
gArguments.output_dir`. If this was not intended, please specify a different
run name by setting the `TrainingArguments.run_name` parameter.
wandb: Using wandb-core as the SDK backend. Please refer to https://wandb.m
e/wandb-core for more information.
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: htt
ps://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/auth
orize
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to
quit:
wandb: ERROR API key must be 40 characters long, yours was 49
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: htt
ps://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/auth
orize
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to
quit:
```

```
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
VBox(children=(Label(value='Waiting for wandb.init()...\r'), FloatProgress(v
alue=0.011112686977778214, max=1.0…
```

Tracking run with wandb version 0.18.3

Run data is saved locally in `/kaggle/working/wandb/run-20241120_155229-vgpn8ib1`

Syncing run **bert-base-cased-sequence-classification** to Weights & Biases (docs)

View project at https://wandb.ai/powisa1142-night/huggingface

View run at https://wandb.ai/powisa1142-night/huggingface/runs/vgpn8ib1

```
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
```

[4500/4500 51:22, Epoch 2/2]

Epoch	Training Loss	Validation Loss	Accuracy
1	0.179200	0.171837	0.942500
2	0.104900	0.162864	0.947583

```
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
```

```
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
```

Out[13]: TrainOutput(global_step=4500, training_loss=0.1619327129787869, metrics={'t
rain_runtime': 3108.2942, 'train_samples_per_second': 69.491, 'train_steps_
per_second': 1.448, 'total_flos': 1.556839626091776e+16, 'train_loss': 0.16
19327129787869, 'epoch': 2.0})

Evaluate on the test partition:

```
In [14]: test_ds = ds['test'].map(
    tokenize,
    batched=True,
    remove_columns=['title', 'description', 'text'],
)
test_ds.to_pandas()
```

Map: 0%| | 0/7600 [00:00<?, ? examples/s]

Out[14]:

	label	input_ids	token_type_ids	attention_mask
0	2	[101, 11284, 1116, 1111, 157, 151, 12966, 1170...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
1	3	[101, 1109, 6398, 1110, 1212, 131, 2307, 7219,...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
2	3	[101, 148, 1183, 119, 1881, 16387, 1116, 4468,...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
3	3	[101, 11689, 15906, 6115, 12056, 1116, 1370, 2...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
4	3	[101, 11917, 8914, 119, 19294, 4206, 1106, 215...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
...	...	...	...	...
7595	0	[101, 5596, 1103, 1362, 5284, 5200, 3234, 1384...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
7596	1	[101, 159, 7874, 1110, 2709, 1114, 13875, 1556...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
7597	1	[101, 16247, 2972, 9178, 2409, 4271, 140, 1418...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
7598	2	[101, 126, 1104, 1893, 8167, 10721, 4420, 1107...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
7599	2	[101, 142, 2064, 4164, 3370, 1154, 13519, 1116...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...

7600 rows × 4 columns

In [15]:
```python
output = trainer.predict(test_ds)
output
```

/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/parallel_apply.py:
79: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please
use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.device(device), torch.cuda.stream(stream), autocast(enable
d=autocast_enabled):
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68:
UserWarning: Was asked to gather along dimension 0, but all input tensors we
re scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '

Out[15]: PredictionOutput(predictions=array([[0.04872653, -4.3446593 , 4.095016 ,
          -0.21087903],
                [-0.13182594, -3.623675 , -2.9268608 , 6.3065596],
                [0.01213044, -3.0985224 , -3.1305702 , 6.0712485],
                ...,
                [-0.8701611 , 6.979305 , -2.5117514 , -3.711188],
                [-1.2446883 , -3.9501538 , 6.1298947 , -1.4984075],
                [-2.444557 , -3.7132833 , 2.2244227 , 2.8555458]],
              dtype=float32), label_ids=array([2, 3, 3, ..., 1, 2, 2]), metrics={'t
          est_loss': 0.17411203682422638, 'test_accuracy': 0.9484210526315789, 'test_
          runtime': 33.4763, 'test_samples_per_second': 227.026, 'test_steps_per_seco
          nd': 4.75})

In [16]:
```python
from sklearn.metrics import classification_report

y_true = output.label_ids
y_pred = np.argmax(output.predictions, axis=-1)
target_names = labels
print(classification_report(y_true, y_pred, target_names=target_names))
```

```
              precision    recall  f1-score   support

       World       0.96      0.96      0.96      1900
      Sports       0.99      0.99      0.99      1900
    Business       0.93      0.91      0.92      1900
    Sci/Tech       0.92      0.93      0.93      1900

    accuracy                           0.95      7600
   macro avg       0.95      0.95      0.95      7600
weighted avg       0.95      0.95      0.95      7600
```

El código anterior genera un pipeline de clasificación para texto, usando BERT, para hacerlo, lo que hace es primero conseguir y preparar los datos, dividiiendolos por entrenamiento, prueba y validación, para que el modelo pueda aprender, despues, los tokeniza usando BERT y genera una capa en el modelo para la clasificacion del texto, con la cual finalmente puede entrenar y evaluar el modelo utilizando los datos antetriores, esta pipeline sirve para integrar, tokenizar y ordenar las tareas de clasificacion de texto