

TD3 : Partie 1

Pour cette partie nous devons reprendre le dataset de France Inter que nous avons utilisé dans certains exercices précédemment. Le but est ici de réaliser un modèle d'entraînement le plus performant possible en utilisant diverses techniques vues en cours, puis de le tester sur des données de validation. Nous avons donc choisi un modèle nommé Camembert, et avons essayé de le fine-tune.

Tout d'abord nous avons entraîné notre modèle sans utiliser le freeze de certaines couches.

```
training_args = TrainingArguments(  
    output_dir="my_awesome_wnut_model",  
    learning_rate=2e-5,  
    per_device_train_batch_size=16,  
    per_device_eval_batch_size=16,  
    num_train_epochs=2,  
    weight_decay=0.01,  
    evaluation_strategy="epoch",  
    save_strategy="epoch",  
    load_best_model_at_end=True,  
    push_to_hub=False,  
)  
  
trainer = Trainer(  
    model=model,  
    args=training_args,  
    train_dataset=dataset_train,  
    eval_dataset=dataset_test,  
    tokenizer=tokenizer,  
    data_collator=data_collator,  
    compute_metrics=compute_metrics,  
    callbacks=[StoreMetricsCallback()]  
)
```

Nous avons de base obtenu de bons résultats au niveau du Loss comme de l'accuracy.

```
{'eval_loss': 0.1072530746459961,  
'eval_precision': 0.0,  
'eval_recall': 0.0,  
'eval_f1': 0.0,  
'eval_accuracy': 0.9620978404583517,  
'eval_runtime': 0.2361,  
'eval_samples_per_second': 787.94,  
'eval_steps_per_second': 50.835,  
'epoch': 2.0}
```

Puis nous avons essayé d'appliquer un freeze des deux premières couches, nous pensons que les résultats ne vont pas être concluants car les deux premières couches sont souvent les plus importantes pour la compréhension globale du langage par le modèle.

```
for name, param in model.base_model.named_parameters():  
    param.requires_grad = False  
  
for name, param in model.base_model.named_parameters():  
    if (  
        any(layer_name in name for layer_name in ["layer.0", "layer.1"])  
        and any(layer_type in name for layer_type in ["weight", "bias"])  
        and "attention" not in name  
    ):  
        param.requires_grad = True
```

Et en effet, les résultats sont moins bons qu'auparavant, bien qu'ils restent bons :

```
{'eval_loss': 0.17233748733997345,  
'eval_precision': 0.0,  
'eval_recall': 0.0,  
'eval_f1': 0.0,  
'eval_accuracy': 0.9612163948876157,  
'eval_runtime': 0.2213,  
'eval_samples_per_second': 840.417,  
'eval_steps_per_second': 54.22,  
'epoch': 2.0}
```

Nous essayons maintenant de freezez toujours deux couches, mais cette fois les deux dernières.

```
for name, param in model.base_model.named_parameters():
    param.requires_grad = False

for name, param in model.base_model.named_parameters():
    if (
        any(layer_name in name for layer_name in ["layer.10", "layer.11"])
        and any(layer_type in name for layer_type in ["weight", "bias"])
        and "attention" not in name
    ):
        param.requires_grad = True
```

✓ 0.0s

Étonnement, les résultats sont encore moins bons qu'avec le freeze des deux premières couches.

```
{'eval_loss': 0.19423992931842804,
 'eval_precision': 0.0,
 'eval_recall': 0.0,
 'eval_f1': 0.0,
 'eval_accuracy': 0.9594535037461437,
 'eval_runtime': 0.2369,
 'eval_samples_per_second': 785.001,
 'eval_steps_per_second': 50.645,
 'epoch': 2.0}
```

Nous restons donc sur le freeze des premières couches même si cela nous paraît contre-intuitif, mais nous passons cette fois-ci à 5 couches freezees.

```
for name, param in model.base_model.named_parameters():
    param.requires_grad = False

for name, param in model.base_model.named_parameters():
    if (
        any(layer_name in name for layer_name in ["layer.0", "layer.1", "layer.2", "layer.3", "layer.4"])
        and any(layer_type in name for layer_type in ["weight", "bias"])
        and "attention" not in name
    ):
        param.requires_grad = True
```

Les résultats ne sont pas satisfaisants, ils ne présentent aucune amélioration par rapport au freeze des deux premières couches, même s'il surperforme notre expérimentation des deux dernières couches.