I- Environment Set Up

In [1]:
```
!pip install transformers --quiet
```

In [2]:
```
!pip3 install torch torchvision torchaudio --quiet
```

In [3]:
```
! pip install -U predibase --quiet
```

```
WARNING: typer 0.12.3 does not provide the extra 'all'
```

In [5]:
```
!pip install torchmetrics --quiet
```

In [6]:
```
import os
from predibase import Predibase, DeploymentConfig

PREDIBASE_API_TOKEN = "YOUR_OWN_PREDIBASE_API_TOKEN"
pb = Predibase(api_token = PREDIBASE_API_TOKEN)
os.environ["PREDIBASE_API_TOKEN"] = PREDIBASE_API_TOKEN
```

```
Connected to Predibase as User(id=7a3c57c6-deec-41df-be47-42e6e197efce, username=jc.kouassi@environnement.gouv.ci)
```

II- Data Preprocessing

In [11]:
```
# Define the column name to modify
column_name = 'split'  # Replace with your column name

# Define the new value
train_new_value = 'train'
val_new_value = 'evaluation'
```

In [13]:
```
import pandas as pd

# Limit to 20k rows
#train_df = pd.read_json("hf://datasets/Infernaught/generation_train/generation_train.jsonl", lines=True)
# Define the Excel file path
excel_file = '1-French-based script - en-bci.xlsx'
sheet_name = 'French-based script - en-bci'
train_df = pd.read_excel(excel_file, sheet_name=sheet_name)
train_df = train_df[train_df[column_name] == train_new_value]
train_df.head(5)
```

Out[13]:

	prompt	completion	split
0	The Baoulé language is an African language fro...	Baoulé anien ô ti sran blé anien koun mô ô f...	train
1	You can find this same page in English, French...	Amoun kwla kangan flouwa boué koungba nga i An...	train
2	Contribute!	Amoun ouka i yolê!	train
3	This is an open test wiki of the Wikimedia Inc...	Flouwa boué nga ô ti klounklo sou Wikimedia In...	train
5	See also Help:Manual for an explanation about ...	Amoun nian Help:Manual ékoun sê amoun kinndê i...	train

In [15]:
```
#test_df = pd.read_json("hf://datasets/Infernaught/generation_test/generation_test.jsonl", lines=True)
test_df = pd.read_excel(excel_file, sheet_name=sheet_name)
test_df = test_df[test_df[column_name] == val_new_value]
test_df.head(5)
```

Out[15]:

	prompt	completion	split
4	If you know this language, you are encouraged ...	Sê amoun si Baoulé anien, e fônvô amoun kpa k...	evaluation
16	Everyone has a right to vote and take part in ...	Sran kwlakwla ô le atin mô ô yé i kloun vote, ...	evaluation
34	The hen knows the dawn, but she waits for the ...	Akôbla ô si aliêndjeinniein, kousou ô nian akô...	evaluation
64	Suppress data from administrators as well as o...	Fa administrateurs moun ôni sran ouflè moun bé...	evaluation
98	Dimensions	Sounzounlè moun	evaluation

In [17]:
```
outdir = "./training_results_dir" # This is where the train_set and intermediate and results files will be saved
os.makedirs(outdir, exist_ok=True)
train_file = f"{outdir}/generation_train.jsonl"
train_df.to_json(train_file, lines=True, orient="records")
```

III- Fine-Tuning

In [ ]:
```
# Upload a dataset
try:
    dataset = pb.datasets.from_file(train_file, name="french-basedscript-en-bci")
except:
    dataset = pb.datasets.get("french-basedscript-en-bci")
```

In [ ]:
```
# get a repo
repo = pb.repos.create(name="french-basedscript-en-bci_training_5epochs", description="Created by evaluation notebook", exists_ok=True)
repo
```

In [ ]:
```
# Create an adapter
adapter = pb.adapters.create(
   config={
       "base_model": "llama-3-1-8b-instruct",
       "epochs": 5,
       "rank": 16,
       "learning_rate": 0.0001,
   },
   dataset=dataset,
   repo=repo,
   description="fine-tuned for french-basedscript en-bci translation task"
)
```

IV- Evaluation

In [19]:
```
# SET YOUR PARAMETERS HERE
adapter_id = "french-basedscript-en-bci_training_5epochs/1" # CHANGE THE VERSION NUMBER HERE TO USE THE CORRECT VERSION IF NEEDED
max_new_tokens = "1024"
tenant_id = "1ccaccc4" # This is your Predibase tenant ID that can be found under "Settings"

requests_filepath = f"{outdir}/requests.jsonl"
results_filepath = f"{outdir}/results.jsonl"
```

IV.1- Monitor Progress

In [21]:
```
# Get adapter, blocking call if training is still in progress
#adapter = pb.adapters.get("news-summarizer-model/1")
adapter_id = "french-basedscript-en-bci_training_5epochs/1"
adapter = pb.adapters.get("french-basedscript-en-bci_training_5epochs/1")
adapter
```

Out[21]:
```
Adapter(repo='french-basedscript-en-bci_training_5epochs', tag=1, checkpoint=None, archived=False, base_model='llama-3-1-8b-instruct', description='Training of French-based script - en-bc
i', artifact_path='6c4d3ca1-976b-4830-861c-6ed68cd29e9d/38cbd395e7da4cbf817a250cb3fc15d8/artifacts/model/model_weights', finetuning_error=None, finetuning_job_uuid='6c4d3ca1-976b-4830-861c-
6ed68cd29e9d')
```

In [ ]:

IV.2- Shared Serverless Endpoints (Free)

In [24]:
```
input_prompt="<s>[INST] The following passage is content from a news report. \n Translate from en to bci: [/INST]"
```

IV.2.1- Base model by calling generate without adapter_id

In [27]:
```
lorax_client = pb.deployments.client("llama-3-1-8b-instruct")
print(lorax_client.generate(input_prompt, max_new_tokens=100).generated_text)
```

```
 The following passage is content from a news report.
The 2023–24 UEFA Champions League is the 69th season of Europe's premier club football tournament organised by the Union of European Football Associations (UEFA). The final will be played at
the Wembley Stadium in London, England. The 2023–24 UEFA Champions League will feature 32 teams from 12 countries. The teams will be divided into eight groups of four teams each. The group st
age will begin on
```

IV.2.2- Our fine-tuned model (with adapter_id)

In [30]:
```
lorax_client = pb.deployments.client("llama-3-1-8b-instruct")
print(lorax_client.generate(input_prompt, adapter_id=adapter_id, max_new_tokens=100).generated_text)
```

```
Ndè kpôlè ndè ô floua boué noun ndè mma sou.
```

In [32]:
```
# Another example
```

In [46]:
```
input_prompt_2="<s>[INST] Where do you come from? [/INST] "
```

In [48]:
```
lorax_client = pb.deployments.client("llama-3-1-8b-instruct")
print(lorax_client.generate(input_prompt_2, adapter_id=adapter_id, max_new_tokens=100).generated_text)
```

```
s>[INST] Amoun fi ni? [/INST]
```

IV.2.3- Evaluation (Performance of Our fine-tuned model on unseen data, zero shoot)

IV.2.3.1- Model Evaluation Metrics

In [50]:
```
from torchmetrics.text.rouge import ROUGEScore

def get_rouge(generated_text, target_text):
    rouge = ROUGEScore()
    return rouge([generated_text], [target_text])["rougeL_fmeasure"]
```

IV.2.3.2- Metric Computing Functions

In [54]:
```
adapter_id = "french-basedscript-en-bci_training_5epochs/1"
def Compute_RougeScore(test_list, target_list):
    #
    score = 0
    total = 0
    for index, test_item in enumerate(test_list):
    #for test_item in test_list:
        #
        lorax_client = pb.deployments.client("llama-3-1-8b-instruct")
        generated_text = lorax_client.generate(test_item, adapter_id=adapter_id, max_new_tokens=100).generated_text
        target_text = target_list[index]
        #print("generated_text : {}".format(generated_text))
        score += get_rouge(generated_text, target_text)
        total += 1
    return f"Rouge Binary accuracy flex score: {score/total}"
```

IV.2.3.3- Evaluation on the Validation set

IV.2.3.3.1- Evaluation - Validation set

In [56]:
```
column_name = 'split'  # Replace with your column name
val_new_value = 'evaluation'

ev_validation_df = pd.read_excel(excel_file, sheet_name=sheet_name)
ev_validation_df = ev_validation_df[ev_validation_df[column_name] == val_new_value]
ev_validation_df = ev_validation_df["prompt"]
ev_validation_df.head(5)
```

Out[56]:
```
4     If you know this language, you are encouraged ...
16    Everyone has a right to vote and take part in ...
34    The hen knows the dawn, but she waits for the ...
64    Suppress data from administrators as well as o...
98                                           Dimensions
Name: prompt, dtype: object
```

IV.2.3.3.2- Evaluation - Target set for the Validation set

In [58]:
```
column_name = 'split'  # Replace with your column name
val_new_value = 'evaluation'

ev_val_target_df = pd.read_excel(excel_file, sheet_name=sheet_name)
ev_val_target_df = ev_val_target_df[ev_val_target_df[column_name] == val_new_value]
ev_val_target_df = ev_val_target_df["completion"]
ev_val_target_df.head(5)
```

Out[58]:
```
4     Sê amoun si Baoulé anien, e fônvô amoun kpa k...
16    Sran kwlakwla ô le atin mô ô yé i kloun vote, ...
34    Akôbla ô si aliêndjeinniein, kousou ô nian akô...
64    Fa administrateurs moun ôni sran ouflè moun bé...
98                                  Sounzounlè moun
Name: completion, dtype: object
```

In [ ]:

IV.2.3.3.3- Scores' Computing

In [60]:
```
ev_validation_list = ev_validation_df.tolist()
ev_val_target_list = ev_val_target_df.tolist()
```

In [62]:
```
Compute_RougeScore(ev_validation_list, ev_val_target_list)
```

Out[62]:
```
'Rouge Binary accuracy flex score: 0.1947985589504242'
```

In [ ]:

IV.2.3.4- Evaluation on a Test set (unseen data - zero shoot)

IV.2.3.4.1- Evaluation Test set

In [68]:
```
excel_file = 'Test sets.xlsx'
sheet_name = 'English test set'
column_name = 'English'
test_df = pd.read_excel(excel_file, sheet_name=sheet_name)
test_df = test_df[column_name]
test_df.head(5)
```

Out[68]:
```
0              We don't go to school on Sundays.
1                       I live in Côte d'Ivoire.
2                    The dove is a symbol of peace.
3    He/She is singing a mockingly allusive song.
4                    A dragonfly is flying outside.
Name: English, dtype: object
```

IV.2.3.4.2- Evaluation Target set for the Test set

In [70]:
```
excel_file = 'Test sets.xlsx'
sheet_name = 'Baoulé French test set'
column_name = 'Baoulé fr'
test_target_df = pd.read_excel(excel_file, sheet_name=sheet_name)
test_target_df = test_target_df[column_name]
test_target_df.head(5)
```

Out[70]:
```
0    Bé kôman souklou Monein tchein moun.
1                N tran Côte d’Ivoire lô.
2            Aublé ti awoundjôê i nzôliê.
3                   Ô sou to asansé djué.
4     Azaanzaan koun sou tou goua sou lô.
Name: Baoulé fr, dtype: object
```

IV.2.3.4.3- Scores' Computing

In [72]:
```
test_list = test_df.tolist()
test_target_list = test_target_df.tolist()
```

In [73]:
```
Compute_RougeScore(test_list, test_target_list)
```

Out[73]:
```
'Rouge Binary accuracy flex score: 0.18897174298763275'
```

IV.2.3.4.4- Score Computing with a simpler text

In [75]:
```
excel_file = 'Test sets.xlsx'
en_sheet_name = 'English test set simple'
bci_sheet_name = 'Baoulé French test set simple'
en_column_name = 'English'
bci_column_name = 'Baoulé fr'

simple_en_test_df = pd.read_excel(excel_file, sheet_name=en_sheet_name)
simple_en_test_df = simple_en_test_df[en_column_name]
simple_en_test_list = simple_en_test_df.tolist()

simple_target_df = pd.read_excel(excel_file, sheet_name=bci_sheet_name)
simple_target_df = simple_target_df[bci_column_name]
simple_target_list = simple_target_df.tolist()
```

In [76]:
```
Compute_RougeScore(simple_en_test_list, simple_target_list)
```

Out[76]:
```
'Rouge Binary accuracy flex score: 0.5851736068725586'
```

In [ ]:

V- Once things are done, the model can be hosted on Predibase (for Inference)

In [ ]:

In [ ]: