INSTRUCTIONS TO FINETUNING WAV2VEC USING THE HF TRANSFORMER PIPELINE APPROACH FOR SPEECH RECOGNITION

1. Installation of relevant libraries in the virtual environment
    a. Pytorch `[pip3 install torch torchvision torchaudio]`
    b. Transformers
    c. Datasets

- Create a virtual environment with the version of Python
- Create venv: **python3 -m venv <your-venv-name>**
- Activate venv: **source ~/<your-venv-name/bin/activate**
- Verify Pytorch and CUDA are installed correctly:

```
python -c "import torch; print(torch.cuda.is_available())"
```

2. Installing HF Transformers

- For the repository to your Github account
- Clone the forked repository to the local disk and add the base repository as a remote:

```
git clone https://github.com/csikasote/transformers.git
cd transformers
pip install -e ".[torch-speech]"
git remote add upstream
https://github.com/huggingface/transformers.git
```

- Create a new branch to hold your development changes.

```
git checkout -b bembaspeech-asr
```

- Set up a Pytorch environment by running the following command in the virtual environment:

```
pip install -e ".[torch-speech]"
```

3. Installing the Dataset library. Simply run the following command:

```
cd ..
git clone https://github.com/huggingface/datasets.git
cd datasets
pip install -e ".[streaming]"
```

```
!pip install transformers[torch]
!pip install accelerate -U
```

4. **[OPTIONAL]**Run the following command in the Python shell to verify that all libraries: transformers and datasets have correctly been installed:

```
from transformers import AutoModelForCTC, AutoProcessor
from datasets import load_dataset

dummy_dataset =
load_dataset("mozilla-foundation/common_voice_11_0", "ab",
split="test")

model =
AutoModelForCTC.from_pretrained("hf-internal-testing/tiny-random-
wav2vec2")
model.to("cuda")

processor =
AutoProcessor.from_pretrained("hf-internal-testing/tiny-random-wa
v2vec2")

input_values = processor(dummy_dataset[0]["audio"]["array"],
return_tensors="pt", sampling_rate=16_000).input_values
input_values = input_values.to("cuda")

logits = model(input_values).logits

assert logits.shape[-1] == 32
```

5. How to fine tuning an Acoustic model using a pretrained [XLS-R](https://huggingface.co/facebook/wav2vec2-xls-r-300m) on the Common Voice dataset. Recommended pre-trained XLS-R checkpoints:
300M parameter version | 1B parameter version | 2B parameter version

- Login into HF:

```
huggingface-cli login
```

- Create your model repository on HF:

```
sudo apt-get install git-lfs
huggingface-cli repo create xls-r-ab-test
git lfs install
git clone https://huggingface.co/hf-test/xls-r-ab-test
```

6. Add the training script and run command to the repository:

- First, copy & paste the official training script from the cloned transformers to the newly created directory:

```
cp
./transformers/examples/pytorch/speech-recognition/run_speech_rec
ognition_ctc.py ./xls-r-ab-test
```

- Next, create a bash file to define the hyper-parameters and configuration for training. More settings can be found here.
- Before training, run the following command to verify that all required libraries are installed:

```
pip install -r
./transformers/examples/pytorch/speech-recognition/requirements.t
xt
```

- Copy the following code snippet in the bash file called **run.sh**

```
echo '''python run_speech_recognition_ctc.py \
 --dataset_name="mozilla-foundation/common_voice_11_0" \
 --model_name_or_path="hf-test/xls-r-dummy" \
 --dataset_config_name="ab" \
 --output_dir="./" \
 --overwrite_output_dir \
 --max_steps="10" \
 --per_device_train_batch_size="2" \
 --learning_rate="3e-4" \
 --save_total_limit="1" \
 --eval_strategy="steps" \
 --text_column_name="sentence" \
 --length_column_name="input_length" \
 --save_steps="5" \
 --layerdrop="0.0" \
 --freeze_feature_encoder \
 --gradient_checkpointing \
 --fp16 \
 --group_by_length \
 --push_to_hub \
 --do_train --do_eval''' > run.sh
```

- Run the following command to start training:

```
bash run.sh
```

7. Evaluating the trained model

- copy the evaluation script, **eval.py,** in the newly created directory :

```
cp
./transformers/examples/research_projects/robust-speech-event/eva
l.py ./xls-r-ab-test
```

```
cd xls-r-ab-test
```

Note: modify some sections of code in the eval.py file as below.

1	`from d` `from datasets import Audio, Dataset, load_dataset` `etric` `import evaluate`

2	`# load metric`
	`# wer = load_metric("wer") # change as below`
	`# cer = load_metric("cer") # chnage as below`
	
	`wer = evaluate.load("wer") # change to this`
	`cer = evaluate.load("cer") # chnage to this`

```
python3 ./eval.py --model_id ./ --dataset
mozilla-foundation/common_voice_11_0 --config ab --split test
--log_outputs
```

References

1. https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/
2. https://github.com/csikasote/transformers/tree/main/examples/research_projects/robust-speech-event
3. https://huggingface.co/blog/wav2vec2-with-ngram