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

1. Installation of relevant libraries in the virtual environment
   1. Pytorch [**pip3 install torch torchvision torchaudio**]
   2. Transformers
   3. 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())”

1. Installing HF Transformers

* For the [repository](https://github.com/huggingface/transformers) 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]"

1. 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

1. [**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-wav2vec2")

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

1. How to fine tuning an Acoustic model using a pretrained [XLS-R](https://huggingface.co/docs/transformers/model_doc/xls_r) on the Common Voice dataset. Recommended pre-trained XLS-R checkpoints:

[300M parameter version](https://huggingface.co/facebook/wav2vec2-xls-r-300m) | [1B parameter version](https://huggingface.co/facebook/wav2vec2-xls-r-1b) | [2B parameter version](https://huggingface.co/facebook/wav2vec2-xls-r-2b)

* 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

1. 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\_recognition\_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](https://github.com/huggingface/transformers/tree/main/examples/pytorch/speech-recognition#connectionist-temporal-classification).
* Before training, run the following command to verify that all required libraries are installed:

pip install -r ./transformers/examples/pytorch/speech-recognition/requirements.txt

* Copy the following code snippet in the bash file called **run.s**h

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

1. Evaluating the trained model

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

cp ./transformers/examples/research\_projects/robust-speech-event/eval.py ./xls-r-ab-test

cd xls-r-ab-test

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



| 1 |  |
| --- | --- |
| 2 |  |

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>