Automatic Speech Recognition

Implementation

HuggingFace ASR Model for Turkish

Turkish has a small dataset on Common Voice so it's mostly used as a starting point to show how we can fine tune a model for a new language.

These are the steps that are followed:

The github repository is at
https://github.com/huggingface/transformers/tree/main/examples/pytorch/speech-recognition

Setup

```
 mkdir asr
 cd asr
 vi run_speech_recognition_seq2seq.py
 vi requirements.txt
 sudo apt-get update
 sudo apt-get install software-properties-common
 sudo add-apt-repository ppa:deadsnakes/ppa
 sudo apt-get update
 sudo apt-get install python3.10
wget https://bootstrap.pypa.io/get-pip.py
python3 get-pip.py
pip --version
pip install virtualenv
virtualenv venv
source venv/bin/activate
pip install  -r requirements.txt
pip install git+https://github.com/huggingface/transformers
git config --global credential.helper store
huggingface-cli login
sudo curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
sudo apt-get install git-lfs
git lfs install
```

Setup for Oracle

```
 mkdir asr
 cd asr
 vi run_speech_recognition_seq2seq.py
 vi requirements.txt
```

```
sudo yum update
sudo yum install epel-release
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo
https://copr.fedorainfracloud.org/coprs/deadsnakes/ppa/repo/epel-7/deadsnakes-ppa-epel-7.repo
sudo yum install -y gcc openssl-devel bzip2-devel libffi-devel zlib-devel
wget https://www.python.org/ftp/python/3.10.0/Python-3.10.0.tgz
tar -xf Python-3.10.0.tgz
cd Python-3.10.0
./configure --enable-optimizations
make -j$(nproc)
sudo make altinstall
sudo ln -s $(pwd)/python /usr/local/bin/python3.10
sudo ln -sf /usr/local/bin/python3.10 /usr/bin/python3
which python3.10
python --version
wget https://bootstrap.pypa.io/get-pip.py
python get-pip.py
pip --version
pip install virtualenv
virtualenv venv
source venv/bin/activate
pip install  -r requirements.txt
sudo yum install git
git --version
sudo yum install git-lfs
pip install git+https://github.com/huggingface/transformers
git config --global credential.helper store
huggingface-cli login
sudo curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.rpm.sh | sudo bash
sudo yum install git-lfs
sudo yum install gcc make ncurses-devel
wget https://ftp.gnu.org/gnu/screen/screen-4.8.0.tar.gz
tar xvf screen-4.8.0.tar.gz
cd screen-4.8.0
./configure
make sudo
make install
```

Train CTC Model for Turkish

```
python run_speech_recognition_ctc.py \
	--dataset_name="common_voice" \
	--model_name_or_path="facebook/wav2vec2-large-xlsr-53" \
	--dataset_config_name="tr" \
	--output_dir="./wav2vec2-common_voice-tr-demo" \
	--overwrite_output_dir \
```

```
--num_train_epochs="15" \
--per_device_train_batch_size="16" \
--gradient_accumulation_steps="2" \
--learning_rate="3e-4" \
--warmup_steps="500" \
--evaluation_strategy="steps" \
--text_column_name="sentence" \
--length_column_name="input_length" \
--save_steps="400" \
--eval_steps="100" \
--layerdrop="0.0" \
--save_total_limit="3" \
--freeze_feature_encoder \
--gradient_checkpointing \
--chars_to_ignore , ? . ! - \; \: \" “ % ‘ ” � \
--fp16 \
--group_by_length \
--push_to_hub \
--do_train --do_eval
```

Final Output

```
{'train_runtime': 6409.9307, 'train_samples_per_second': 8.139, 'train_steps_per_second': 0.255, 'train_loss': 1.070046563688039, 'epoch': 15.0}
100%|████████████████████████████████████████████████████████████████████████████████████████████████████| 1635/1635 [1:46:49<00:00,  3.92s/it]
Saving model checkpoint to ./wav2vec2-common_voice-tr-demo
Configuration saved in ./wav2vec2-common_voice-tr-demo/config.json
Model weights saved in ./wav2vec2-common_voice-tr-demo/pytorch_model.bin
Feature extractor saved in ./wav2vec2-common_voice-tr-demo/preprocessor_config.json
Saving model checkpoint to ./wav2vec2-common_voice-tr-demo
Configuration saved in ./wav2vec2-common_voice-tr-demo/config.json
Model weights saved in ./wav2vec2-common_voice-tr-demo/pytorch_model.bin
Feature extractor saved in ./wav2vec2-common_voice-tr-demo/preprocessor_config.json
Several commits (2) will be pushed upstream.
04/07/2023 22:04:36 - WARNING - huggingface_hub.repository - Several commits (2) will be pushed upstream.
The progress bars may be unreliable.
04/07/2023 22:04:36 - WARNING - huggingface_hub.repository - The progress bars may be unreliable.
Upload file pytorch_model.bin: 1.18GB [04:52, 6.66MB/s]To
https://huggingface.co/tarunanand/wav2vec2-common_voice-tr-demo
   29c21b3..7c2047e  main -> main
04/07/2023 22:09:32 - WARNING - huggingface_hub.repository - To
https://huggingface.co/tarunanand/wav2vec2-common_voice-tr-demo
   29c21b3..7c2047e  main -> main
Upload file pytorch_model.bin:
100%|████████████████████████████████████████████████████████████████████████████████████████████████| 1.18G/1.18G [04:54<00:00, 4.29MB/s]
To https://huggingface.co/tarunanand/wav2vec2-common_voice-tr-demo
   7c2047e..d28c03f  main -> main
04/07/2023 22:09:37 - WARNING - huggingface_hub.repository - To
https://huggingface.co/tarunanand/wav2vec2-common_voice-tr-demo
```

```
   7c2047e..d28c03f  main -> main
***** train metrics *****
  epoch                    =       15.0
  train_loss               =       1.07
  train_runtime            = 1:46:49.93
  train_samples            =       3478
  train_samples_per_second =      8.139
  train_steps_per_second   =      0.255
04/07/2023 22:09:40 - INFO - __main__ - *** Evaluate ***
The following columns in the evaluation set don't have a corresponding argument in `Wav2Vec2ForCTC.forward` and have been ignored: input_length. If input_length are not expected by `Wav2Vec2ForCTC.forward`,  you can safely ignore this message.
***** Running Evaluation *****
  Num examples = 1647
  Batch size = 8
100%|████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████| 206/206 [01:46<00:00,  1.93it/s]
***** eval metrics *****
  epoch                   =       15.0
  eval_loss               =      0.376
  eval_runtime            = 0:01:47.42
  eval_samples            =       1647
  eval_samples_per_second =     15.332
  eval_steps_per_second   =      1.918
  eval_wer                =     0.3521
Saving model checkpoint to ./wav2vec2-common_voice-tr-demo
Configuration saved in ./wav2vec2-common_voice-tr-demo/config.json
Model weights saved in ./wav2vec2-common_voice-tr-demo/pytorch_model.bin
Feature extractor saved in ./wav2vec2-common_voice-tr-demo/preprocessor_config.json
To https://huggingface.co/tarunanand/wav2vec2-common_voice-tr-demo
   d28c03f..781e65f  main -> main
04/07/2023 22:11:39 - WARNING - huggingface_hub.repository - To
https://huggingface.co/tarunanand/wav2vec2-common_voice-tr-demo
   d28c03f..781e65f  main -> main
To https://huggingface.co/tarunanand/wav2vec2-common_voice-tr-demo
   781e65f..e985695  main -> main
04/07/2023 22:11:43 - WARNING - huggingface_hub.repository - To
https://huggingface.co/tarunanand/wav2vec2-common_voice-tr-demo
   781e65f..e985695  main -> main
```

Train CTC Model for Hindi

For Hindi use the following:

Single GPU

```
python run_speech_recognition_ctc.py \
	--dataset_name="common_voice" \
	--model_name_or_path="facebook/wav2vec2-large-xlsr-53" \
	--dataset_config_name="hi" \
	--output_dir="./wav2vec2-common_voice-hi-demo" \
	--overwrite_output_dir \
```

```
--num_train_epochs="15" \
--per_device_train_batch_size="16" \
--gradient_accumulation_steps="2" \
--learning_rate="3e-4" \
--warmup_steps="500" \
--evaluation_strategy="steps" \
--text_column_name="sentence" \
--length_column_name="input_length" \
--save_steps="400" \
--eval_steps="100" \
--layerdrop="0.0" \
--save_total_limit="3" \
--freeze_feature_encoder \
--gradient_checkpointing \
--chars_to_ignore , ? . ! - \; \: \" " % ' " � \
--fp16 \
--group_by_length \
--push_to_hub \
--do_train --do_eval
```

Train Seq2Seq Model for Hindi

Multi GPU Training for Whisper Medium 2xV100-16-240GB_Ubuntu22-14

```
screen -dmS asr torchrun --nproc_per_node 2 run_speech_recognition_seq2seq.py
--model_name_or_path="openai/whisper-medium" --dataset_name="mozilla-foundation/common_voice_11_0"
--dataset_config_name="hi" --language="hindi" --train_split_name="train+validation" --eval_split_name="test"
--max_steps="5000" --output_dir="./whisper-medium-hi" --per_device_train_batch_size="16"
--per_device_eval_batch_size="16" --logging_steps="25" --learning_rate="1e-5" --warmup_steps="500"
--evaluation_strategy="steps" --eval_steps="1000" --save_strategy="steps" --save_steps="1000"
--generation_max_length="225" --preprocessing_num_workers="16" --length_column_name="input_length"
--max_duration_in_seconds="30" --text_column_name="sentence" --freeze_feature_encoder="False"
--gradient_checkpointing --group_by_length --fp16 --overwrite_output_dir --do_train --do_eval
--predict_with_generate --use_auth_token --push_to_hub
```

Multi GPU Training for Whisper Large V2 GDC-2xA100-32-230GB_Ubuntu22-175

```
screen -dmS asr torchrun --nproc_per_node 2 run_speech_recognition_seq2seq.py
--model_name_or_path="openai/whisper-large-v2" --dataset_name="mozilla-foundation/common_voice_11_0"
--dataset_config_name="hi" --language="hindi" --train_split_name="train+validation" --eval_split_name="test"
--max_steps="5000" --output_dir="./whisper-large-v2-hi" --per_device_train_batch_size="10"
--per_device_eval_batch_size="10" --logging_steps="25" --learning_rate="1e-5" --warmup_steps="500"
--evaluation_strategy="steps" --eval_steps="1000" --save_strategy="steps" --save_steps="1000"
--generation_max_length="225" --preprocessing_num_workers="16" --length_column_name="input_length"
--max_duration_in_seconds="30" --text_column_name="sentence" --freeze_feature_encoder="False"
--gradient_checkpointing --group_by_length --fp16 --overwrite_output_dir --do_train --do_eval
--predict_with_generate --use_auth_token --push_to_hub
```

Train Seq2Seq Model for Kangri

Multi GPU Training for Whisper Large V2 GDC-2xA100-32-230GB_Ubuntu22-175 on Snow-Mountain Dataset

```
screen -dmS asr torchrun    --nproc_per_node 2 run_speech_recognition_seq2seq.py
--model_name_or_path="vasista22/whisper-hindi-large-v2"
--dataset_name="bridgeconn/snow-mountain"   --dataset_config_name="kangri"
--language="hindi"  --train_split_name="train_500"     --eval_split_name="test_common"
--max_steps="5000"  --output_dir="./whisper-large-v2-kangri"
--per_device_train_batch_size="8"   --per_device_eval_batch_size="8"
--logging_steps="25"    --learning_rate="1e-5"  --warmup_steps="500"
--evaluation_strategy="steps"   --eval_steps="1000"     --save_strategy="steps"
--save_steps="1000"     --generation_max_length="225"
--preprocessing_num_workers="16"    --length_column_name="input_length"
--max_duration_in_seconds="30"  --text_column_name="sentence"
--freeze_feature_encoder="False"    --gradient_checkpointing    --group_by_length
--fp16  --overwrite_output_dir  --do_train  --do_eval   --predict_with_generate
--use_auth_token    --push_to_hub
```

Troubleshooting

1. ProcessGroupNCCL is only supported with GPUs, no GPUs found!
   This issue is usually due to the Cloud Provider not having proper drivers for CUDA. Contact IT/Support.
2. pytorch_cuda_alloc_conf cuda out of memory - Tried the following and the suggestions in the link below but ultimately had to go with a lower model.
   export CUDA_VISIBLE_DEVICES=0,1
   export PYTORCH_CUDA_ALLOC_CONF=garbage_collection_threshold:0.6,max_split_size_mb:128

   https://medium.com/@snk.nitin/how-to-solve-cuda-out-of-memory-error-850bb247cfb2
3.

Command Line Arguments

python3 run_speech_recognition_seq2seq.py --help

```
(venv) root@localhost:~/asr# python3 run_speech_recognition_seq2seq.py --help
usage: run_speech_recognition_seq2seq.py [-h] --model_name_or_path

MODEL_NAME_OR_PATH [--config_name CONFIG_NAME] [--tokenizer_name
TOKENIZER_NAME] [--feature_extractor_name FEATURE_EXTRACTOR_NAME]
                        [--cache_dir CACHE_DIR] [--use_fast_tokenizer
[USE_FAST_TOKENIZER]] [--no_use_fast_tokenizer] [--model_revision
MODEL_REVISION] [--use_auth_token [USE_AUTH_TOKEN]]
                        [--freeze_feature_encoder [FREEZE_FEATURE_ENCODER]]
[--no_freeze_feature_encoder] [--freeze_encoder [FREEZE_ENCODER]]
                        [--forced_decoder_ids FORCED_DECODER_IDS
[FORCED_DECODER_IDS ...]] [--suppress_tokens SUPPRESS_TOKENS
[SUPPRESS_TOKENS ...]] [--apply_spec_augment [APPLY_SPEC_AUGMENT]]
                        [--dataset_name DATASET_NAME] [--dataset_config_name
DATASET_CONFIG_NAME] [--overwrite_cache [OVERWRITE_CACHE]]
[--preprocessing_num_workers PREPROCESSING_NUM_WORKERS]
                        [--max_train_samples MAX_TRAIN_SAMPLES]
[--max_eval_samples MAX_EVAL_SAMPLES] [--audio_column_name
AUDIO_COLUMN_NAME] [--text_column_name TEXT_COLUMN_NAME]
                        [--max_duration_in_seconds
MAX_DURATION_IN_SECONDS] [--min_duration_in_seconds
MIN_DURATION_IN_SECONDS] [--preprocessing_only [PREPROCESSING_ONLY]]
                        [--train_split_name TRAIN_SPLIT_NAME] [--eval_split_name
EVAL_SPLIT_NAME] [--do_lower_case [DO_LOWER_CASE]] [--no_do_lower_case]
[--language LANGUAGE] [--task TASK] --output_dir
                        OUTPUT_DIR [--overwrite_output_dir
[OVERWRITE_OUTPUT_DIR]] [--do_train [DO_TRAIN]] [--do_eval [DO_EVAL]]
[--do_predict [DO_PREDICT]] [--evaluation_strategy {no,steps,epoch}]
                        [--prediction_loss_only [PREDICTION_LOSS_ONLY]]
[--per_device_train_batch_size PER_DEVICE_TRAIN_BATCH_SIZE]
[--per_device_eval_batch_size PER_DEVICE_EVAL_BATCH_SIZE]
                        [--per_gpu_train_batch_size
PER_GPU_TRAIN_BATCH_SIZE] [--per_gpu_eval_batch_size
PER_GPU_EVAL_BATCH_SIZE] [--gradient_accumulation_steps
GRADIENT_ACCUMULATION_STEPS]
                        [--eval_accumulation_steps EVAL_ACCUMULATION_STEPS]
[--eval_delay EVAL_DELAY] [--learning_rate LEARNING_RATE] [--weight_decay
WEIGHT_DECAY] [--adam_beta1 ADAM_BETA1]
```

[--adam_beta2 ADAM_BETA2] [--adam_epsilon
ADAM_EPSILON] [--max_grad_norm MAX_GRAD_NORM] [--num_train_epochs
NUM_TRAIN_EPOCHS] [--max_steps MAX_STEPS]
[--lr_scheduler_type
{linear,cosine,cosine_with_restarts,polynomial,constant,constant_with_warmup,inverse_sqrt}] [--warmup_ratio WARMUP_RATIO] [--warmup_steps WARMUP_STEPS]
[--log_level {debug,info,warning,error,critical,passive}]
[--log_level_replica {debug,info,warning,error,critical,passive}] [--log_on_each_node
[LOG_ON_EACH_NODE]]
[--no_log_on_each_node] [--logging_dir LOGGING_DIR]
[--logging_strategy {no,steps,epoch}] [--logging_first_step [LOGGING_FIRST_STEP]]
[--logging_steps LOGGING_STEPS]
[--logging_nan_inf_filter [LOGGING_NAN_INF_FILTER]]
[--no_logging_nan_inf_filter] [--save_strategy {no,steps,epoch}] [--save_steps
SAVE_STEPS]
[--save_total_limit SAVE_TOTAL_LIMIT] [--save_safetensors
[SAVE_SAFETENSORS]] [--save_on_each_node [SAVE_ON_EACH_NODE]]
[--no_cuda [NO_CUDA]] [--use_mps_device [USE_MPS_DEVICE]]
[--seed SEED] [--data_seed DATA_SEED] [--jit_mode_eval
[JIT_MODE_EVAL]] [--use_ipex [USE_IPEX]] [--bf16 [BF16]] [--fp16 [FP16]]
[--fp16_opt_level FP16_OPT_LEVEL]
[--half_precision_backend {auto,cuda_amp,apex,cpu_amp}]
[--bf16_full_eval [BF16_FULL_EVAL]] [--fp16_full_eval [FP16_FULL_EVAL]] [--tf32
TF32] [--local_rank LOCAL_RANK]
[--xpu_backend {mpi,ccl,gloo}] [--tpu_num_cores
TPU_NUM_CORES] [--tpu_metrics_debug [TPU_METRICS_DEBUG]] [--debug
DEBUG] [--dataloader_drop_last [DATALOADER_DROP_LAST]]
[--eval_steps EVAL_STEPS] [--dataloader_num_workers
DATALOADER_NUM_WORKERS] [--past_index PAST_INDEX] [--run_name
RUN_NAME] [--disable_tqdm DISABLE_TQDM]
[--remove_unused_columns
[REMOVE_UNUSED_COLUMNS]] [--no_remove_unused_columns] [--label_names
LABEL_NAMES [LABEL_NAMES ...]] [--load_best_model_at_end
[LOAD_BEST_MODEL_AT_END]]
[--metric_for_best_model METRIC_FOR_BEST_MODEL]
[--greater_is_better GREATER_IS_BETTER] [--ignore_data_skip
[IGNORE_DATA_SKIP]] [--sharded_ddp SHARDED_DDP] [--fsdp FSDP]
[--fsdp_min_num_params FSDP_MIN_NUM_PARAMS]
[--fsdp_config FSDP_CONFIG] [--fsdp_transformer_layer_cls_to_wrap
FSDP_TRANSFORMER_LAYER_CLS_TO_WRAP] [--deepspeed DEEPSPEED]

```
                              [--label_smoothing_factor LABEL_SMOOTHING_FACTOR]
                              [--optim
{adamw_hf,adamw_torch,adamw_torch_fused,adamw_torch_xla,adamw_apex_fused,adafactor,adamw_bnb_8bit,adamw_anyprecision,sgd,adagrad}] [--optim_args
OPTIM_ARGS]
                              [--adafactor [ADAFACTOR]] [--group_by_length
[GROUP_BY_LENGTH]] [--length_column_name LENGTH_COLUMN_NAME]
[--report_to REPORT_TO [REPORT_TO ...]]
                              [--ddp_find_unused_parameters
DDP_FIND_UNUSED_PARAMETERS] [--ddp_bucket_cap_mb
DDP_BUCKET_CAP_MB] [--dataloader_pin_memory [DATALOADER_PIN_MEMORY]]
[--no_dataloader_pin_memory]
                              [--skip_memory_metrics [SKIP_MEMORY_METRICS]]
[--no_skip_memory_metrics] [--use_legacy_prediction_loop
[USE_LEGACY_PREDICTION_LOOP]] [--push_to_hub [PUSH_TO_HUB]]
                              [--resume_from_checkpoint
RESUME_FROM_CHECKPOINT] [--hub_model_id HUB_MODEL_ID] [--hub_strategy
{end,every_save,checkpoint,all_checkpoints}] [--hub_token HUB_TOKEN]
                              [--hub_private_repo [HUB_PRIVATE_REPO]]
[--gradient_checkpointing [GRADIENT_CHECKPOINTING]]
[--include_inputs_for_metrics [INCLUDE_INPUTS_FOR_METRICS]]
                              [--fp16_backend {auto,cuda_amp,apex,cpu_amp}]
[--push_to_hub_model_id PUSH_TO_HUB_MODEL_ID] [--push_to_hub_organization
PUSH_TO_HUB_ORGANIZATION]
                              [--push_to_hub_token PUSH_TO_HUB_TOKEN]
[--mp_parameters MP_PARAMETERS] [--auto_find_batch_size
[AUTO_FIND_BATCH_SIZE]] [--full_determinism [FULL_DETERMINISM]]
                              [--torchdynamo TORCHDYNAMO] [--ray_scope
RAY_SCOPE] [--ddp_timeout DDP_TIMEOUT] [--torch_compile [TORCH_COMPILE]]
[--torch_compile_backend TORCH_COMPILE_BACKEND]
                              [--torch_compile_mode TORCH_COMPILE_MODE]
[--sortish_sampler [SORTISH_SAMPLER]] [--predict_with_generate
[PREDICT_WITH_GENERATE]] [--generation_max_length
GENERATION_MAX_LENGTH]
                              [--generation_num_beams GENERATION_NUM_BEAMS]
[--generation_config GENERATION_CONFIG]
options:
  -h, --help            show this help message and exit
  --model_name_or_path MODEL_NAME_OR_PATH
```

Path to pretrained model or model identifier from huggingface.co/models (default: None)
--config_name CONFIG_NAME
Pretrained config name or path if not the same as model_name (default: None)
--tokenizer_name TOKENIZER_NAME
Pretrained tokenizer name or path if not the same as model_name (default: None)
--feature_extractor_name FEATURE_EXTRACTOR_NAME
feature extractor name or path if not the same as model_name (default: None)
--cache_dir CACHE_DIR
Where to store the pretrained models downloaded from huggingface.co (default: None)
--use_fast_tokenizer [USE_FAST_TOKENIZER]
Whether to use one of the fast tokenizer (backed by the tokenizers library) or not. (default: True)
--no_use_fast_tokenizer
Whether to use one of the fast tokenizer (backed by the tokenizers library) or not. (default: False)
--model_revision MODEL_REVISION
The specific model version to use (can be a branch name, tag name or commit id). (default: main)
--use_auth_token [USE_AUTH_TOKEN]
Will use the token generated when running `huggingface-cli login` (necessary to use this script with private models). (default: False)
--freeze_feature_encoder [FREEZE_FEATURE_ENCODER]
Whether to freeze the feature encoder layers of the model. (default: True)
--no_freeze_feature_encoder
Whether to freeze the feature encoder layers of the model. (default: False)
--freeze_encoder [FREEZE_ENCODER]
Whether to freeze the entire encoder of the seq2seq model. (default: False)
--forced_decoder_ids FORCED_DECODER_IDS [FORCED_DECODER_IDS ...]
A list of pairs of integers which indicates a mapping from generation indices to token indices that will be forced before sampling. For example, [[0, 123]] means the first generated token will
always be a token of index 123. (default: None)

--suppress_tokens SUPPRESS_TOKENS [SUPPRESS_TOKENS ...]
A list of tokens that will be suppressed at generation. (default: None)
--apply_spec_augment [APPLY_SPEC_AUGMENT]
Whether to apply *SpecAugment* data augmentation to the input features. This is currently only relevant for Wav2Vec2, HuBERT, WavLM and Whisper models. (default: False)
--dataset_name DATASET_NAME
The name of the dataset to use (via the datasets library). (default: None)
--dataset_config_name DATASET_CONFIG_NAME
The configuration name of the dataset to use (via the datasets library). (default: None)
--overwrite_cache [OVERWRITE_CACHE]
Overwrite the cached training and evaluation sets (default: False)
--preprocessing_num_workers PREPROCESSING_NUM_WORKERS
The number of processes to use for the preprocessing. (default: None)
--max_train_samples MAX_TRAIN_SAMPLES
For debugging purposes or quicker training, truncate the number of training examples to this value if set. (default: None)
--max_eval_samples MAX_EVAL_SAMPLES
For debugging purposes or quicker training, truncate the number of evaluation examples to this value if set. (default: None)
--audio_column_name AUDIO_COLUMN_NAME
The name of the dataset column containing the audio data. Defaults to 'audio' (default: audio)
--text_column_name TEXT_COLUMN_NAME
The name of the dataset column containing the text data. Defaults to 'text' (default: text)
--max_duration_in_seconds MAX_DURATION_IN_SECONDS
Truncate audio files that are longer than `max_duration_in_seconds` seconds to 'max_duration_in_seconds` (default: 20.0)
--min_duration_in_seconds MIN_DURATION_IN_SECONDS
Filter audio files that are shorter than `min_duration_in_seconds` seconds (default: 0.0)
--preprocessing_only [PREPROCESSING_ONLY]
Whether to only do data preprocessing and skip training. This is especially useful when data preprocessing errors out in distributed training due to timeout. In this case, one should run the
preprocessing in a non-distributed setup with `preprocessing_only=True` so that the cached datasets can consequently be loaded in distributed training (default: False)

```
  --train_split_name TRAIN_SPLIT_NAME
                        The name of the training data set split to use (via the datasets library).
Defaults to 'train' (default: train)
  --eval_split_name EVAL_SPLIT_NAME
                        The name of the training data set split to use (via the datasets library).
Defaults to 'train' (default: test)
  --do_lower_case [DO_LOWER_CASE]
                        Whether the target text should be lower cased. (default: True)
  --no_do_lower_case    Whether the target text should be lower cased. (default: False)
  --language LANGUAGE   Language for multilingual fine-tuning. This argument should
be set for multilingual fine-tuning only. For English speech recognition, it should be set
to `None`. (default: None)
  --task TASK           Task, either `transcribe` for speech recognition or `translate` for
speech translation. (default: transcribe)
  --output_dir OUTPUT_DIR
                        The output directory where the model predictions and checkpoints will be
written. (default: None)
  --overwrite_output_dir [OVERWRITE_OUTPUT_DIR]
                        Overwrite the content of the output directory. Use this to continue training
if output_dir points to a checkpoint directory. (default: False)
  --do_train [DO_TRAIN]
                        Whether to run training. (default: False)
  --do_eval [DO_EVAL]   Whether to run eval on the dev set. (default: False)
  --do_predict [DO_PREDICT]
                        Whether to run predictions on the test set. (default: False)
  --evaluation_strategy {no,steps,epoch}
                        The evaluation strategy to use. (default: no)
  --prediction_loss_only [PREDICTION_LOSS_ONLY]
                        When performing evaluation and predictions, only returns the loss.
(default: False)
  --per_device_train_batch_size PER_DEVICE_TRAIN_BATCH_SIZE
                        Batch size per GPU/TPU core/CPU for training. (default: 8)
  --per_device_eval_batch_size PER_DEVICE_EVAL_BATCH_SIZE
                        Batch size per GPU/TPU core/CPU for evaluation. (default: 8)
  --per_gpu_train_batch_size PER_GPU_TRAIN_BATCH_SIZE
                        Deprecated, the use of `--per_device_train_batch_size` is preferred.
Batch size per GPU/TPU core/CPU for training. (default: None)
  --per_gpu_eval_batch_size PER_GPU_EVAL_BATCH_SIZE
                        Deprecated, the use of `--per_device_eval_batch_size` is preferred.
Batch size per GPU/TPU core/CPU for evaluation. (default: None)
```

```
  --gradient_accumulation_steps GRADIENT_ACCUMULATION_STEPS
                        Number of updates steps to accumulate before performing a
backward/update pass. (default: 1)
  --eval_accumulation_steps EVAL_ACCUMULATION_STEPS
                        Number of predictions steps to accumulate before moving the tensors to
the CPU. (default: None)
  --eval_delay EVAL_DELAY
                        Number of epochs or steps to wait for before the first evaluation can be
performed, depending on the evaluation_strategy. (default: 0)
  --learning_rate LEARNING_RATE
                        The initial learning rate for AdamW. (default: 5e-05)
  --weight_decay WEIGHT_DECAY
                        Weight decay for AdamW if we apply some. (default: 0.0)
  --adam_beta1 ADAM_BETA1
                        Beta1 for AdamW optimizer (default: 0.9)
  --adam_beta2 ADAM_BETA2
                        Beta2 for AdamW optimizer (default: 0.999)
  --adam_epsilon ADAM_EPSILON
                        Epsilon for AdamW optimizer. (default: 1e-08)
  --max_grad_norm MAX_GRAD_NORM
                        Max gradient norm. (default: 1.0)
  --num_train_epochs NUM_TRAIN_EPOCHS
                        Total number of training epochs to perform. (default: 3.0)
  --max_steps MAX_STEPS
                        If > 0: set total number of training steps to perform. Override
num_train_epochs. (default: -1)
  --lr_scheduler_type
{linear,cosine,cosine_with_restarts,polynomial,constant,constant_with_warmup,inverse_
sqrt}
                        The scheduler type to use. (default: linear)
  --warmup_ratio WARMUP_RATIO
                        Linear warmup over warmup_ratio fraction of total steps. (default: 0.0)
  --warmup_steps WARMUP_STEPS
                        Linear warmup over warmup_steps. (default: 0)
  --log_level {debug,info,warning,error,critical,passive}
                        Logger log level to use on the main node. Possible choices are the log
levels as strings: 'debug', 'info', 'warning', 'error' and 'critical', plus a 'passive' level which
doesn't set anything
                        and lets the application set the level. Defaults to 'passive'. (default:
passive)
```

```
  --log_level_replica {debug,info,warning,error,critical,passive}
                        Logger log level to use on replica nodes. Same choices and defaults as
``log_level`` (default: warning)
  --log_on_each_node [LOG_ON_EACH_NODE]
                        When doing a multinode distributed training, whether to log once per
node or just once on the main node. (default: True)
  --no_log_on_each_node
                        When doing a multinode distributed training, whether to log once per
node or just once on the main node. (default: False)
  --logging_dir LOGGING_DIR
                        Tensorboard log dir. (default: None)
  --logging_strategy {no,steps,epoch}
                        The logging strategy to use. (default: steps)
  --logging_first_step [LOGGING_FIRST_STEP]
                        Log the first global_step (default: False)
  --logging_steps LOGGING_STEPS
                        Log every X updates steps. (default: 500)
  --logging_nan_inf_filter [LOGGING_NAN_INF_FILTER]
                        Filter nan and inf losses for logging. (default: True)
  --no_logging_nan_inf_filter
                        Filter nan and inf losses for logging. (default: False)
  --save_strategy {no,steps,epoch}
                        The checkpoint save strategy to use. (default: steps)
  --save_steps SAVE_STEPS
                        Save checkpoint every X updates steps. (default: 500)
  --save_total_limit SAVE_TOTAL_LIMIT
                        Limit the total amount of checkpoints. Deletes the older checkpoints in
the output_dir. Default is unlimited checkpoints (default: None)
  --save_safetensors [SAVE_SAFETENSORS]
                        Use safetensors saving and loading for state dicts instead of default
torch.load and torch.save. (default: False)
  --save_on_each_node [SAVE_ON_EACH_NODE]
                        When doing multi-node distributed training, whether to save models and
checkpoints on each node, or only on the main one (default: False)
  --no_cuda [NO_CUDA]   Do not use CUDA even when it is available (default: False)
  --use_mps_device [USE_MPS_DEVICE]
                        Whether to use Apple Silicon chip based `mps` device. (default: False)
  --seed SEED           Random seed that will be set at the beginning of training. (default:
42)
  --data_seed DATA_SEED
```

Random seed to be used with data samplers. (default: None)
--jit_mode_eval [JIT_MODE_EVAL]
Whether or not to use PyTorch jit trace for inference (default: False)
--use_ipex [USE_IPEX]
Use Intel extension for PyTorch when it is available, installation: 'https://github.com/intel/intel-extension-for-pytorch' (default: False)
--bf16 [BF16] Whether to use bf16 (mixed) precision instead of 32-bit. Requires Ampere or higher NVIDIA architecture or using CPU (no_cuda). This is an experimental API and it may change. (default: False)
--fp16 [FP16] Whether to use fp16 (mixed) precision instead of 32-bit (default: False)
--fp16_opt_level FP16_OPT_LEVEL
For fp16: Apex AMP optimization level selected in ['O0', 'O1', 'O2', and 'O3']. See details at https://nvidia.github.io/apex/amp.html (default: O1)
--half_precision_backend {auto,cuda_amp,apex,cpu_amp}
The backend to be used for half precision. (default: auto)
--bf16_full_eval [BF16_FULL_EVAL]
Whether to use full bfloat16 evaluation instead of 32-bit. This is an experimental API and it may change. (default: False)
--fp16_full_eval [FP16_FULL_EVAL]
Whether to use full float16 evaluation instead of 32-bit (default: False)
--tf32 TF32 Whether to enable tf32 mode, available in Ampere and newer GPU architectures. This is an experimental API and it may change. (default: None)
--local_rank LOCAL_RANK
For distributed training: local_rank (default: -1)
--xpu_backend {mpi,ccl,gloo}
The backend to be used for distributed training on Intel XPU. (default: None)
--tpu_num_cores TPU_NUM_CORES
TPU: Number of TPU cores (automatically passed by launcher script) (default: None)
--tpu_metrics_debug [TPU_METRICS_DEBUG]
Deprecated, the use of `--debug tpu_metrics_debug` is preferred. TPU: Whether to print debug metrics (default: False)
--debug DEBUG Whether or not to enable debug mode. Current options: `underflow_overflow` (Detect underflow and overflow in activations and weights), `tpu_metrics_debug` (print debug metrics on TPU). (default:
)
--dataloader_drop_last [DATALOADER_DROP_LAST]

Drop the last incomplete batch if it is not divisible by the batch size. (default: False)
--eval_steps EVAL_STEPS
Run an evaluation every X steps. (default: None)
--dataloader_num_workers DATALOADER_NUM_WORKERS
Number of subprocesses to use for data loading (PyTorch only). 0 means that the data will be loaded in the main process. (default: 0)
--past_index PAST_INDEX
If >=0, uses the corresponding part of the output as the past state for next step. (default: -1)
--run_name RUN_NAME An optional descriptor for the run. Notably used for wandb logging. (default: None)
--disable_tqdm DISABLE_TQDM
Whether or not to disable the tqdm progress bars. (default: None)
--remove_unused_columns [REMOVE_UNUSED_COLUMNS]
Remove columns not required by the model when using an nlp.Dataset. (default: True)
--no_remove_unused_columns
Remove columns not required by the model when using an nlp.Dataset. (default: False)
--label_names LABEL_NAMES [LABEL_NAMES ...]
The list of keys in your dictionary of inputs that correspond to the labels. (default: None)
--load_best_model_at_end [LOAD_BEST_MODEL_AT_END]
Whether or not to load the best model found during training at the end of training. (default: False)
--metric_for_best_model METRIC_FOR_BEST_MODEL
The metric to use to compare two different models. (default: None)
--greater_is_better GREATER_IS_BETTER
Whether the `metric_for_best_model` should be maximized or not. (default: None)
--ignore_data_skip [IGNORE_DATA_SKIP]
When resuming training, whether or not to skip the first epochs and batches to get to the same training data. (default: False)
--sharded_ddp SHARDED_DDP
Whether or not to use sharded DDP training (in distributed training only). The base option should be `simple`, `zero_dp_2` or `zero_dp_3` and you can add CPU-offload to `zero_dp_2` or

`zero_dp_3` like this: zero_dp_2 offload` or `zero_dp_3 offload`. You can add auto-wrap to `zero_dp_2` or `zero_dp_3` with the same syntax: zero_dp_2 auto_wrap` or `zero_dp_3 auto_wrap`.
(default:)
--fsdp FSDP Whether or not to use PyTorch Fully Sharded Data Parallel (FSDP) training (in distributed training only). The base option should be `full_shard`, `shard_grad_op` or `no_shard` and you can add
CPU-offload to `full_shard` or `shard_grad_op` like this: full_shard offload` or `shard_grad_op offload`. You can add auto-wrap to `full_shard` or `shard_grad_op` with the same syntax:
full_shard auto_wrap` or `shard_grad_op auto_wrap`. (default:)
--fsdp_min_num_params FSDP_MIN_NUM_PARAMS
This parameter is deprecated. FSDP's minimum number of parameters for Default Auto Wrapping. (useful only when `fsdp` field is passed). (default: 0)
--fsdp_config FSDP_CONFIG
Config to be used with FSDP (Pytorch Fully Sharded Data Parallel). The value is either afsdp json config file (e.g., `fsdp_config.json`) or an already loaded json file as `dict`. (default: None)
--fsdp_transformer_layer_cls_to_wrap FSDP_TRANSFORMER_LAYER_CLS_TO_WRAP
This parameter is deprecated. Transformer layer class name (case-sensitive) to wrap, e.g, `BertLayer`, `GPTJBlock`, `T5Block` .... (useful only when `fsdp` flag is passed). (default: None)
--deepspeed DEEPSPEED
Enable deepspeed and pass the path to deepspeed json config file (e.g. ds_config.json) or an already loaded json file as a dict (default: None)
--label_smoothing_factor LABEL_SMOOTHING_FACTOR
The label smoothing epsilon to apply (zero means no label smoothing). (default: 0.0)
--optim {adamw_hf,adamw_torch,adamw_torch_fused,adamw_torch_xla,adamw_apex_fused,adafactor,adamw_bnb_8bit,adamw_anyprecision,sgd,adagrad}
The optimizer to use. (default: adamw_hf)
--optim_args OPTIM_ARGS
Optional arguments to supply to optimizer. (default: None)
--adafactor [ADAFACTOR]
Whether or not to replace AdamW by Adafactor. (default: False)
--group_by_length [GROUP_BY_LENGTH]
Whether or not to group samples of roughly the same length together when batching. (default: False)

--length_column_name LENGTH_COLUMN_NAME
Column name with precomputed lengths to use when grouping by length. (default: length)
--report_to REPORT_TO [REPORT_TO ...]
The list of integrations to report the results and logs to. (default: None)
--ddp_find_unused_parameters DDP_FIND_UNUSED_PARAMETERS
When using distributed training, the value of the flag `find_unused_parameters` passed to `DistributedDataParallel`. (default: None)
--ddp_bucket_cap_mb DDP_BUCKET_CAP_MB
When using distributed training, the value of the flag `bucket_cap_mb` passed to `DistributedDataParallel`. (default: None)
--dataloader_pin_memory [DATALOADER_PIN_MEMORY]
Whether or not to pin memory for DataLoader. (default: True)
--no_dataloader_pin_memory
Whether or not to pin memory for DataLoader. (default: False)
--skip_memory_metrics [SKIP_MEMORY_METRICS]
Whether or not to skip adding of memory profiler reports to metrics. (default: True)
--no_skip_memory_metrics
Whether or not to skip adding of memory profiler reports to metrics. (default: False)
--use_legacy_prediction_loop [USE_LEGACY_PREDICTION_LOOP]
Whether or not to use the legacy prediction_loop in the Trainer. (default: False)
--push_to_hub [PUSH_TO_HUB]
Whether or not to upload the trained model to the model hub after training. (default: False)
--resume_from_checkpoint RESUME_FROM_CHECKPOINT
The path to a folder with a valid checkpoint for your model. (default: None)
--hub_model_id HUB_MODEL_ID
The name of the repository to keep in sync with the local `output_dir`. (default: None)
--hub_strategy {end,every_save,checkpoint,all_checkpoints}
The hub strategy to use when `--push_to_hub` is activated. (default: every_save)
--hub_token HUB_TOKEN
The token to use to push to the Model Hub. (default: None)
--hub_private_repo [HUB_PRIVATE_REPO]
Whether the model repository is private or not. (default: False)

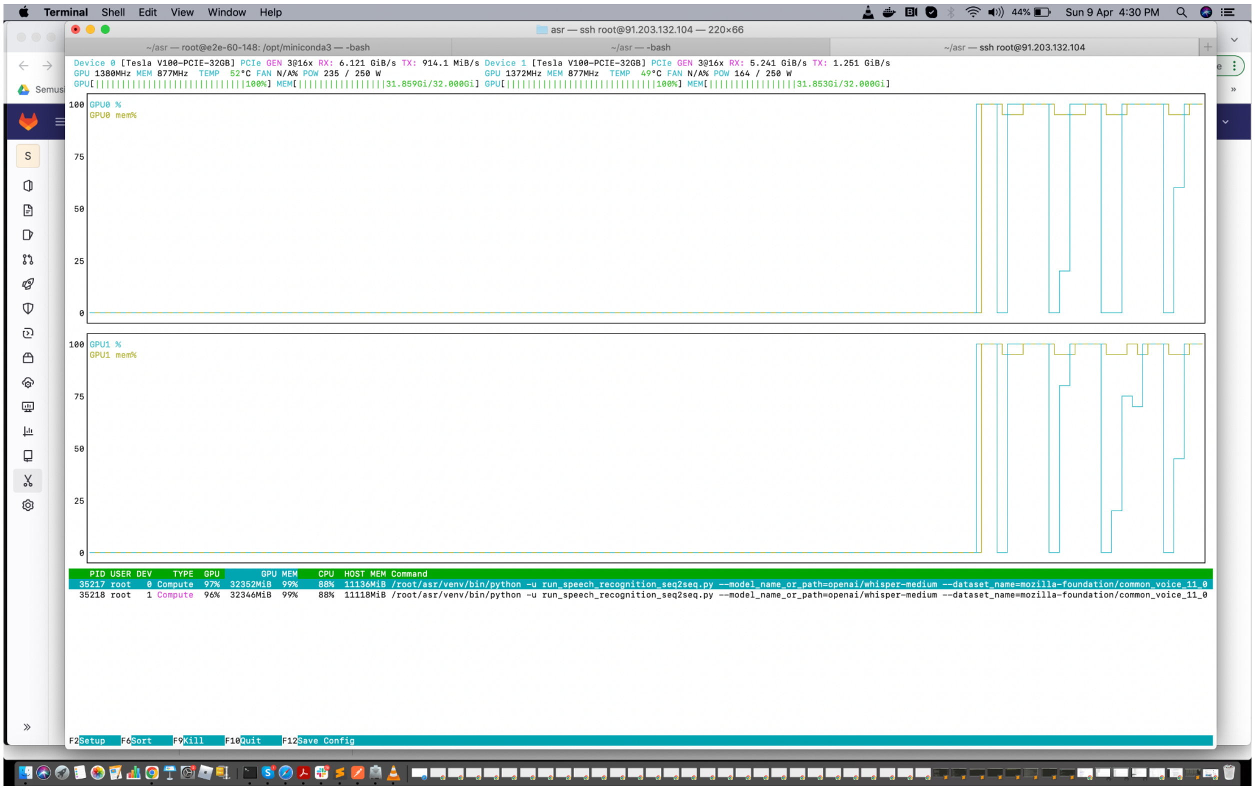
--gradient_checkpointing [GRADIENT_CHECKPOINTING]
If True, use gradient checkpointing to save memory at the expense of slower backward pass. (default: False)
--include_inputs_for_metrics [INCLUDE_INPUTS_FOR_METRICS]
Whether or not the inputs will be passed to the `compute_metrics` function. (default: False)
--fp16_backend {auto,cuda_amp,apex,cpu_amp}
Deprecated. Use half_precision_backend instead (default: auto)
--push_to_hub_model_id PUSH_TO_HUB_MODEL_ID
The name of the repository to which push the `Trainer`. (default: None)
--push_to_hub_organization PUSH_TO_HUB_ORGANIZATION
The name of the organization in with to which push the `Trainer`. (default: None)
--push_to_hub_token PUSH_TO_HUB_TOKEN
The token to use to push to the Model Hub. (default: None)
--mp_parameters MP_PARAMETERS
Used by the SageMaker launcher to send mp-specific args. Ignored in Trainer (default:)
--auto_find_batch_size [AUTO_FIND_BATCH_SIZE]
Whether to automatically decrease the batch size in half and rerun the training loop again each time a CUDA Out-of-Memory was reached (default: False)
--full_determinism [FULL_DETERMINISM]
Whether to call enable_full_determinism instead of set_seed for reproducibility in distributed training. Important: this will negatively impact the performance, so only use it for debugging.
(default: False)
--torchdynamo TORCHDYNAMO
This argument is deprecated, use `--torch_compile_backend` instead. (default: None)
--ray_scope RAY_SCOPE
The scope to use when doing hyperparameter search with Ray. By default, `"last"` will be used. Ray will then use the last checkpoint of all trials, compare those, and select the best one.
However, other options are also available. See the Ray documentation (https://docs.ray.io/en/latest/tune/api_docs/analysis.html#ray.tune.ExperimentAnalysis.get_best_trial) for more options.
(default: last)
--ddp_timeout DDP_TIMEOUT
Overrides the default timeout for distributed training (value should be given in seconds). (default: 1800)

```
  --torch_compile [TORCH_COMPILE]
                        If set to `True`, the model will be wrapped in `torch.compile`. (default:
False)
  --torch_compile_backend TORCH_COMPILE_BACKEND
                        Which backend to use with `torch.compile`, passing one will trigger a
model compilation. (default: None)
  --torch_compile_mode TORCH_COMPILE_MODE
                        Which mode to use with `torch.compile`, passing one will trigger a model
compilation. (default: None)
  --sortish_sampler [SORTISH_SAMPLER]
                        Whether to use SortishSampler or not. (default: False)
  --predict_with_generate [PREDICT_WITH_GENERATE]
                        Whether to use generate to calculate generative metrics (ROUGE,
BLEU). (default: False)
  --generation_max_length GENERATION_MAX_LENGTH
                        The `max_length` to use on each evaluation loop when
`predict_with_generate=True`. Will default to the `max_length` value of the model
configuration. (default: None)
  --generation_num_beams GENERATION_NUM_BEAMS
                        The `num_beams` to use on each evaluation loop when
`predict_with_generate=True`. Will default to the `num_beams` value of the model
configuration. (default: None)
  --generation_config GENERATION_CONFIG
                        Model id, file path or url pointing to a GenerationConfig json file, to use
during prediction. (default: None)
```

System Monitoring

Monitoring the GPU of Intel, nvidia - we can use a tool like htop

apt install nvtop



Setup References

[Installing Python, Pip and VirtualEnv]
[Setup environment variables in Linux]
[Install Transformers from source]
[Create tokens for HuggingFace CLI]

References

Kangri & Low Resource Language ASR

Automatic Recognition of Dialects of Himachal Pradesh Using MFCC &GMM
https://ieeexplore.ieee.org/document/8988336/figures#figures

2022-01-13 - Automatic Speech Recognition for Low Resource Languages – Satwinder Singh
https://www.youtube.com/watch?v=tHBtGBS60vA

Generic ASR

Speech Papers with Code for Google Fleurs
https://paperswithcode.com/dataset/fleurs

Speech-to-Text request construction
https://cloud.google.com/speech-to-text/docs/speech-to-text-requests

gcloud auth activate-service-account
https://cloud.google.com/sdk/gcloud/reference/auth/activate-service-account

Method: speech.longrunningrecognize
https://cloud.google.com/speech-to-text/docs/reference/rest/v1/speech/longrunningrecognize#TranscriptOutputConfig

Send a recognition request with model adaptation
https://cloud.google.com/speech-to-text/docs/adaptation

Method: projects.locations.phraseSets.create
https://cloud.google.com/speech-to-text/docs/reference/rest/v1/projects.locations.phraseSets/create

Improve transcription results with model adaptation
https://cloud.google.com/speech-to-text/docs/adaptation-model#whats_next

oAuth2 Playground
https://developers.google.com/oauthplayground/?code=4/0ARtbsJp8pdfKKYHfdDD_nGOGb1GKQ1FkFCdNMHDIHtrNvYaiAkE5_XPZKkWRPKu88JIK2A&scope=https://www.googleapis.com/auth/cloud-platform

gcloud auth application-default print-access-token
https://cloud.google.com/sdk/gcloud/reference/auth/application-default/print-access-token

Authentication
https://googleapis.dev/python/google-api-core/latest/auth.html

My GCP Project
https://console.cloud.google.com/apis/credentials?authuser=4&project=warm-airline-366511&pli=1

Execute code samples
https://developers.google.com/explorer-help/code-samples

Install the Google Cloud CLI
https://cloud.google.com/sdk/docs/install-sdk

Transcribe long audio files into text
https://cloud.google.com/speech-to-text/docs/async-recognize#speech_transcribe_async_gcs-protocol

M4A to WAV Converter
https://cloudconvert.com/m4a-to-wav

How Big of a Deal Is 'Whisper' for ASR and Multilingual Transcription?
https://slator.com/how-big-a-deal-is-whisper-for-asr-multilingual-transcription/

How to create a speech dataset for ASR, TTS, and other speech tasks
https://ogunlao.github.io/blog/2021/01/26/how-to-create-speech-dataset.html

Asr label data
https://www.google.com/search?q=asr+label+data&rlz=1C5CHFA_enIN855AE858&oq=asr+label+data&aqs=chrome..69i57j33i160l3.5844j0j7&sourceid=chrome&ie=UTF-8

Installing Whisper
https://colab.research.google.com/github/openai/whisper/blob/master/notebooks/LibriSpeech.ipynb#scrollTo=v5hvo8QWN-a9

https://usfoor.com/nvidia-riva-sets-new-bar-for-fully-customizable-speech-ai/

Benchmarking OpenAI Whisper on non-English datasets
https://blog.deepgram.com/benchmarking-openai-whisper-for-non-english-asr/

Fine Tuning Whisper
https://huggingface.co/docs/transformers/main/en/model_doc/whisper#transformers.WhisperForConditionalGeneration
https://huggingface.co/blog/fine-tune-whisper
https://huggingface.co/spaces/openai/whisper/discussions/6

Troubleshooting Whisper/HuggingFace Fine Tuning
https://discuss.huggingface.co/t/trainer-runtimeerror-the-size-of-tensor-a-462-must-match-the-size-of-tensor-b-448-at-non-singleton-dimension-1/26010

Hindi ASR

1. https://kunal-dhawan.weebly.com/asr-system-for-hindi-language-from-scratch.html
   (Used Kaldi - which is in CPP)
2. https://blog.deepgram.com/6-challenges-asr-hindi/
   Challenges we might face
3. https://ohmvikrant.github.io/Hindi-ASR/, https://github.com/OhmVikrant/ASR-for-Hindi
   Kaldi ASR - better version, 98% Accuracy
4. https://www.researchgate.net/publication/260508111_Development_and_Suitability_of_Indian_Languages_Speech_Database_for_Building_Watson_Based_ASR_System
   Using WATSON
5. https://docs.nvidia.com/deeplearning/riva/user-guide/docs/tutorials/New-language-adaptation/Hindi/README.html
   NVIDIA Deep Learning
6. https://sites.google.com/view/asr-challenge/leaderboard
   IIT Hyderabad 7% WER
7. https://huggingface.co/speechbrain/asr-whisper-large-v2-commonvoice-hi
8. https://ai4bharat.org/, https://ai4bharat.org/indicwav2vec
   ai4Bharat - very cool stuff, lots of data
9. https://huggingface.co/skylord/wav2vec2-large-xlsr-hindi
10. https://www.cse.iitd.ac.in/~aseth/Gram_Vaani_ASR_Challenge_Interspeech.pdf

Punjabi ASR

11. https://ohmvikrant.github.io/Punjabi-ASR/
    Kaldi ASR with very low WER
12. https://www.youtube.com/watch?v=tHBtGBS60vA&ab_channel=InstituteofDataScience%28IDS%29%2CNUS

Datasets

Hindi

1. https://officechai.com/stories/indian-govt-releases-version-of-openais-whisper-model-which-turns-hindi-speech-into-text/
2. https://www.twine.net/blog/top-indian-language-datasets/
3. https://data.ldcil.org/hindi-raw-speech-corpus
4. https://ai4bharat.org/shrutilipi
5. GramVaani ASR Corpus https://www.openslr.org/118/
6. ULCA ASR Corpus https://github.com/Open-Speech-EkStep/ULCA-asr-dataset-corpus
   https://github.com/Open-Speech-EkStep/ULCA-asr-dataset-corpus/blob/main/LICENSE

7. Google/Fleurs https://huggingface.co/datasets/google/fleurs/viewer/hi_in/train https://huggingface.co/datasets/google/xtreme_s

Dataset References

1. https://huggingface.co/docs/datasets/audio_dataset
2. https://huggingface.co/docs/datasets/upload_dataset

Testing Data for Whisper Medium Small Fine Tuned on Hindi

1. Predictions - https://drive.google.com/file/d/1vNGeifOx0pRpxyIWiymYs0gh6NMgXZox/view?usp=sharing
2. References - https://drive.google.com/file/d/1regaxf53W64S7GI7uq_UlpODMACR3tD3/view?usp=sharing
3. **WER - 65% - Zero shot.**