**Setting up the Python Environment:**

conda create --name spapl\_kidsasr python=3.9.7

in venv:

pip install torch transformers==4.32.1 accelerate==0.20.3 datasets soundfile librosa numpy evaluate dataclasses-json jiwer pygments

add to path (if warning):

nano ~/.bashrc

add this line: export PATH=$PATH:/home/klp65/.local/bin

source ~/.bashrc

**Kaldi Installation:**

a) Clone Kaldi repository

git clone <https://github.com/kaldi-asr/kaldi.git>

instructions in <https://github.com/kaldi-asr/kaldi/blob/master/INSTALL>

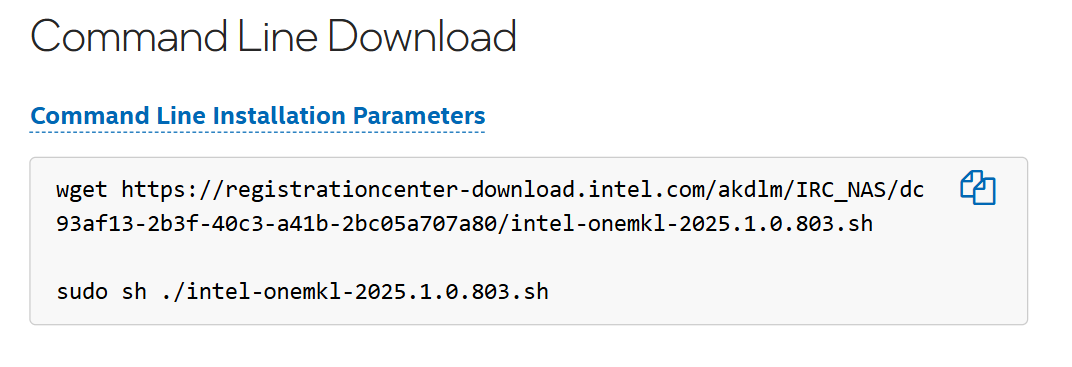
b) Build the tools (including sclite) – instructions in <https://github.com/kaldi-asr/kaldi/blob/master/tools/INSTALL>

cd kaldi/tools

extras/check\_dependencies.sh

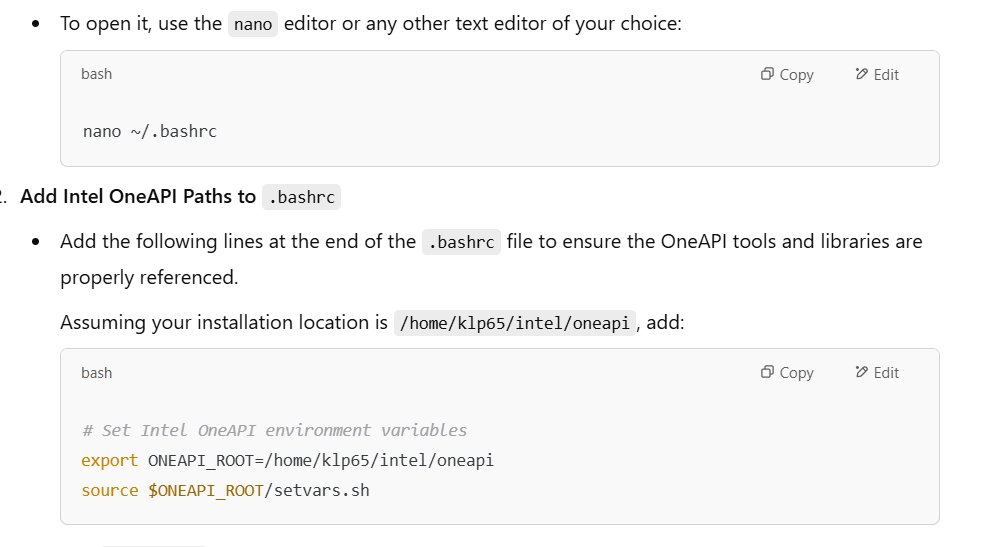
reported missing MKL:

<https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl-download.html?operatingsystem=linux&linux-install=online>

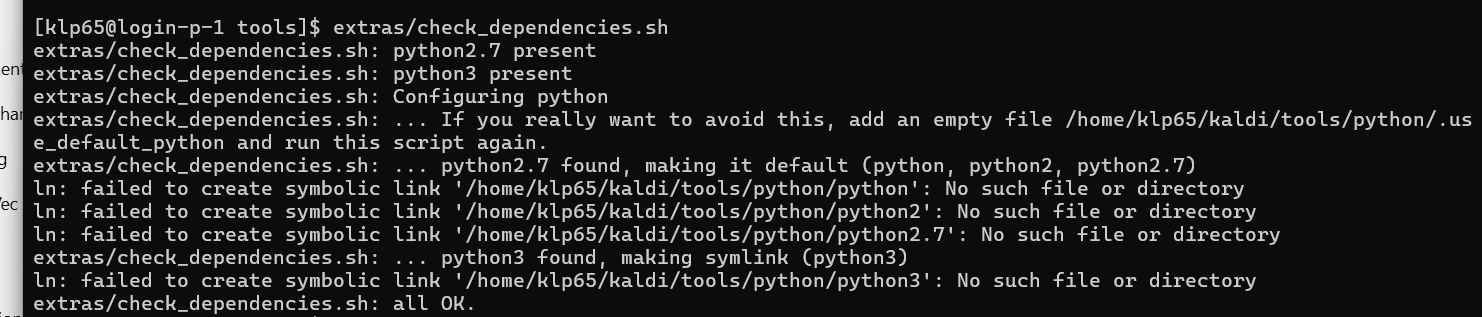


Installation location will show after successful installation. No need for sudo.

After you’re done, check installation location and update bashrc and do source ~/.bashrc after



Make sure all ok:



Openfst issue: (downgrade to 1.6.7 instead of 1.8.4)

cd /home/klp65/kaldi/tools

wget http://www.openfst.org/twiki/pub/FST/FstDownload/openfst-1.6.7.tar.gz

tar -xzf openfst-1.6.7.tar.gz

cd openfst-1.6.7

./configure --prefix=/home/klp65/kaldi/tools/openfst --enable-shared --enable-ngram-fsts --enable-lookahead-fsts LDFLAGS="-pthread"

make -j4

cd ..

**in Makefile change the OPENFST VERSION:**

****

after that: in tools

make clean

make

\*\* remember to “make clean” before “make” if you changed compiler

Ensure GCC is being used:

export CC=gcc

export CXX=g++

c) build the source -- [follow instructions in https://github.com/kaldi-asr/kaldi/blob/master/src/INSTALL](https://github.com/kaldi-asr/kaldi/blob/master/src/INSTALL)

cd ../src

./configure --shared

make depend -j 8

make -j 8

**include PATH for sclite:**

(spapl\_kidsasr) [klp65@login-q-1 OGI]$ find /home/klp65/kaldi -name "sclite"

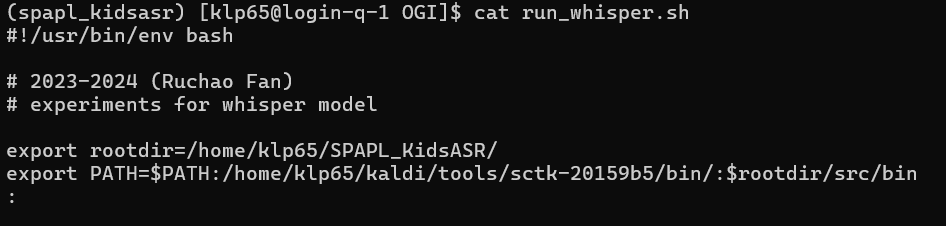
/home/klp65/kaldi/tools/sctk-20159b5/bin/sclite

/home/klp65/kaldi/tools/sctk-20159b5/src/sclite

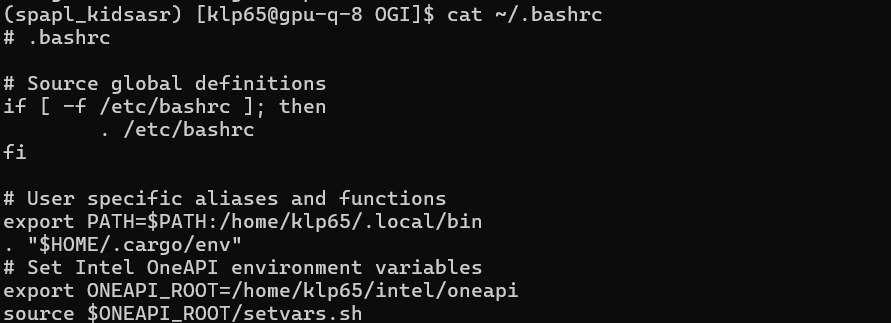
/home/klp65/kaldi/tools/sctk-20159b5/src/sclite/sclite

(spapl\_kidsasr) [klp65@login-q-1 OGI]$ export PATH=$PATH:/home/klp65/kaldi/tools/sctk-20159b5/bin

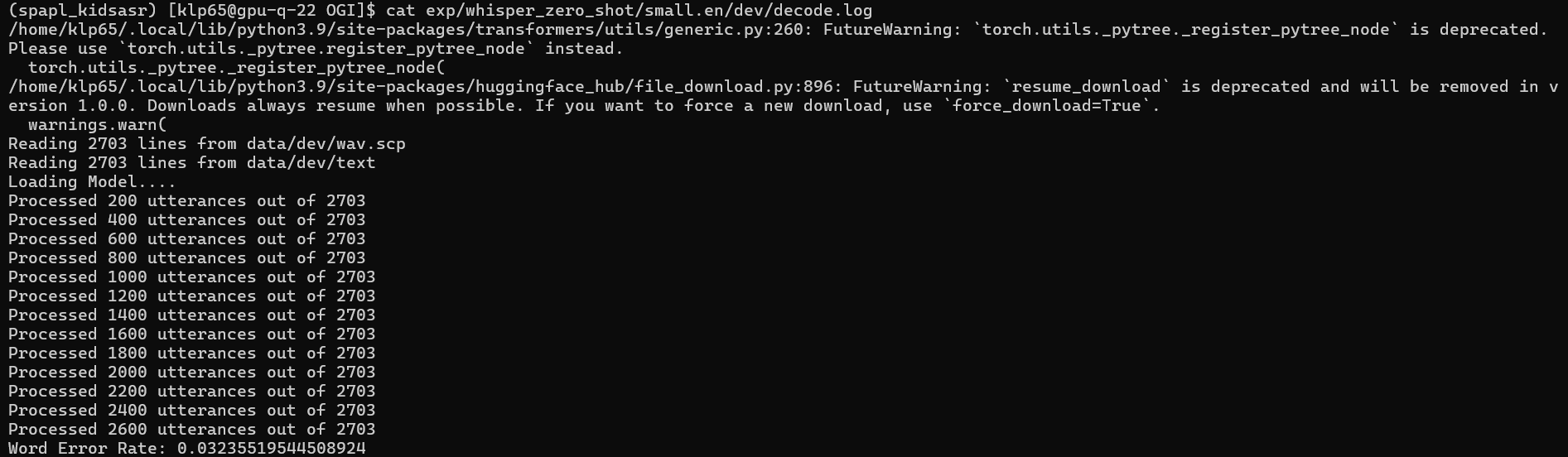
**Change paths at the top for run\_whisper.sh:**

****

**Bashrc file:**

****

*cat exp/whisper\_zero\_shot/small.en/dev/decode.log to debug asr\_decode.py*

**

**Data format in OGI Kids:**

<https://github.com/OSU-slatelab/OGI-kids-phoneme-recognition/blob/main/ogi_prepare.py>

.wav files are in:

{data\_folder}/speech/scripted/

transcription files are in:

with open(train\_align\_file) as f:

train\_alignments = json.load(f)

with open(valid\_align\_file) as f:

valid\_alignments = json.load(f)

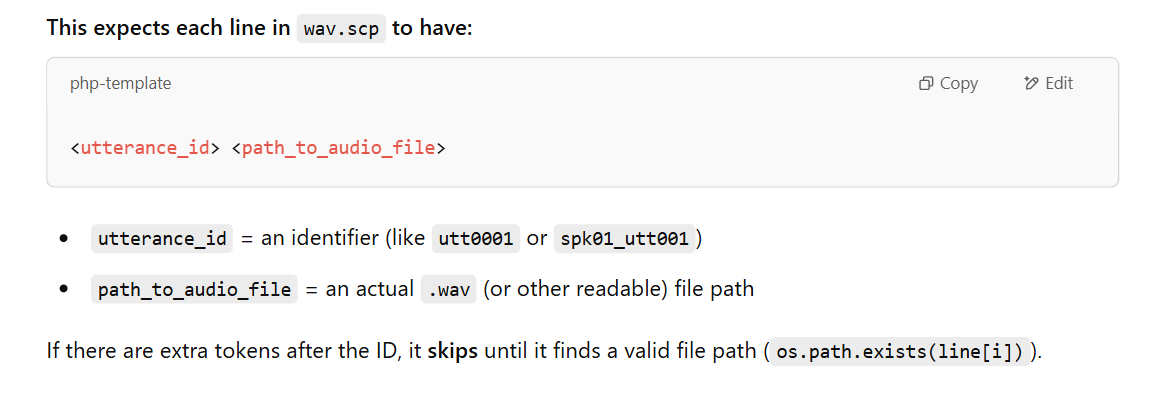
JSON files contain alignments —> probably mappings from sentence ID → transcribed words

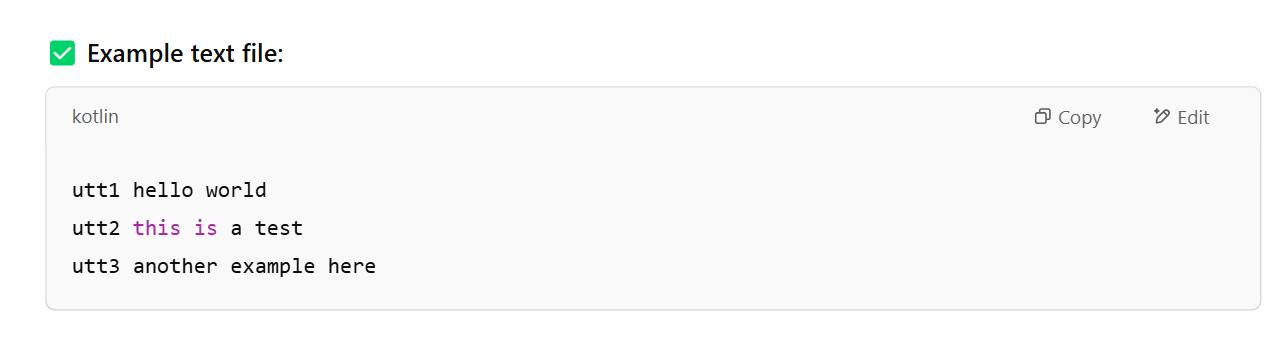
**Data accepted in code:**

wav.scp file: connects every utterance (sentence said by one person during particular recording session) with an audio file related to this utterance.

/src/data/whisper\_loader.py: loads dataset for Whisper, does data augmentation based on arguments in whisper\_small\_train.yaml

From whisper\_loader.py:





**For OGI dataset:**

Stage 1/3: Evaluation of Whisper (in run\_whisper.sh)

/egs/OGI/data/dev/wav.scp

/egs/OGI/data/dev/text

/egs/OGI/data/test/wav.scp

/egs/OGI/data/test/text

/egs/OGI/data/spont\_al/wav.scp

/egs/OGI/data/spont\_al/text

Stage 2: Finetuning of Whisper (in whisper\_small\_train.yaml)

Training data:

/egs/OGI/data/train/wav.scp

/egs/OGI/data/train/text

Validation data:

/egs/OGI/data/dev/wav.scp

/egs/OGI/data/dev/text

tasks, both OGI and MyST:

1. stage 1: evaluation of baseline whisper model

2. run stage 2 and 3: full-finetuning without data augmentation

3. run stage 2 and 3: full-finetuning with VTLP

4. run stage 2 and 3: full-finetuning with SP

5. run stage 2 and 3: full-finetuning with PP

6. run stage 2 and 3: full-finetuning with SA

try combinations:

1. SA + PP

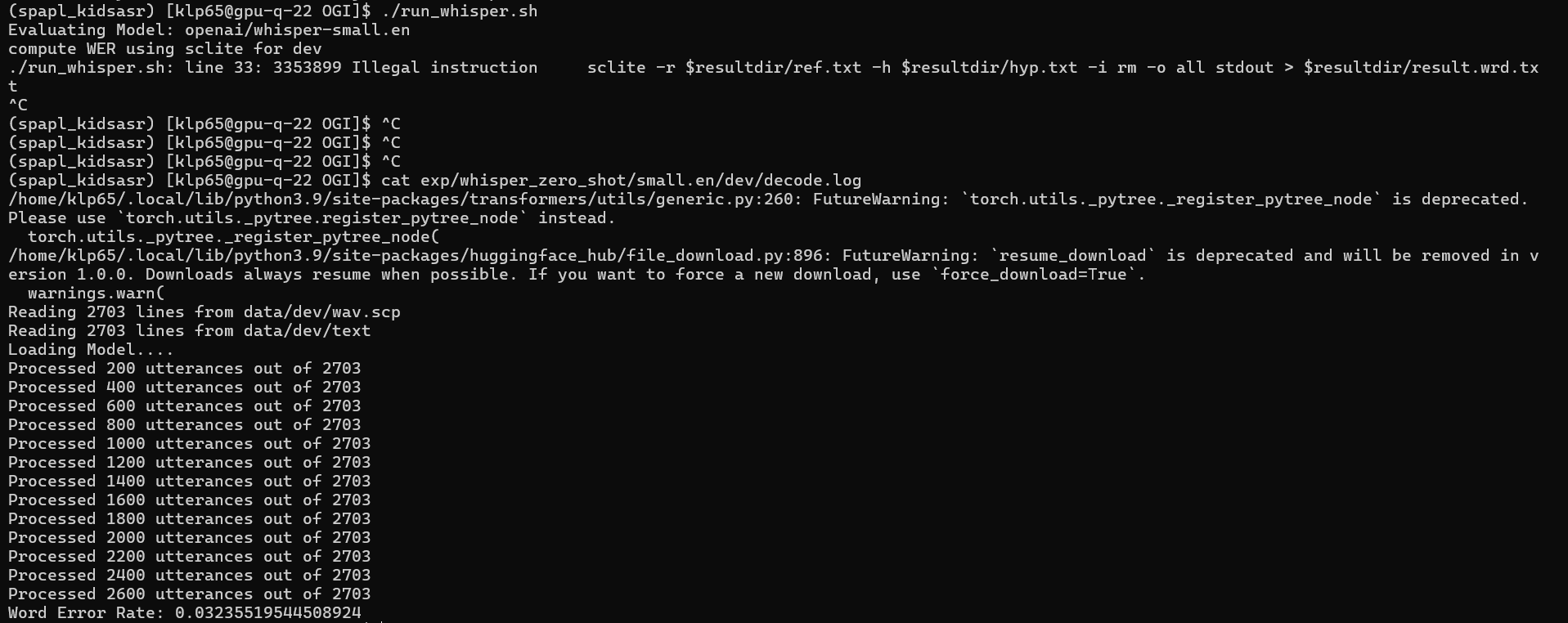
2. SA + SP

3. SA + VTLP

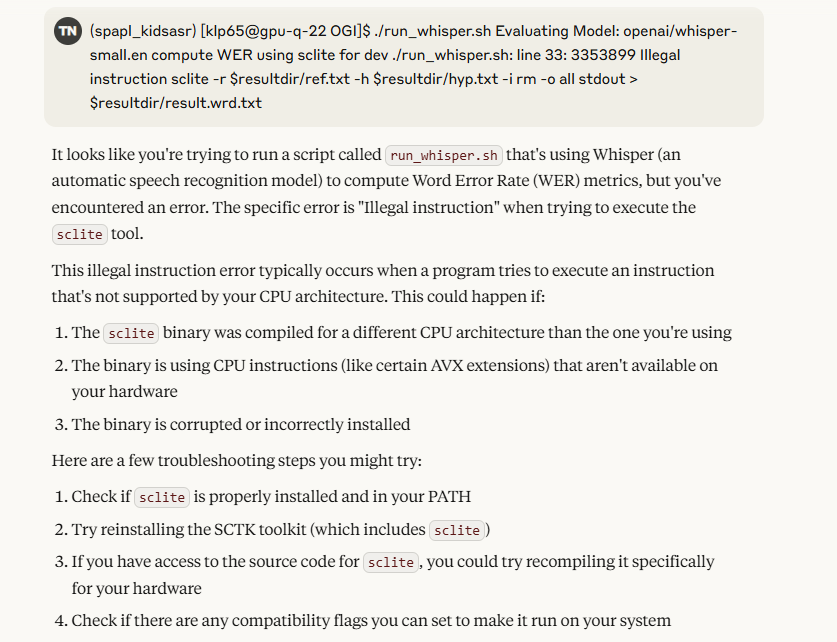
try PIF with VLTP and PP.

**Testing OGI corpus code**

**Current Issues for Stage 1:**

****

Sclite incompatibility but WER is recorded in decode.log file.



**Issues for Stage 2 (No DA, No PEFT):**

*For debugging: ./run\_whisper.sh 2>&1 | tee logfile.txt (this prints output in terminal and saves in logfile.txt)*

To navigate to end of the file: press G

**Issue 1:**

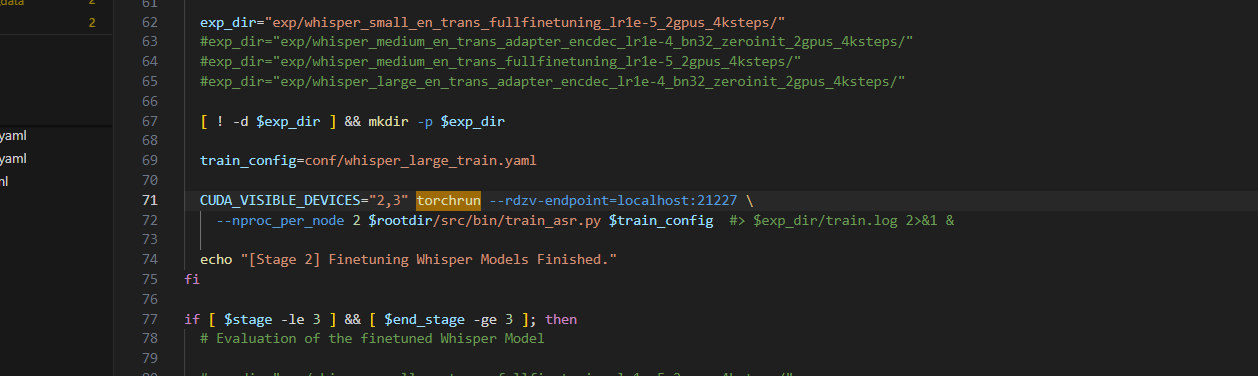
[rank0]: ValueError: FP16 Mixed precision training with AMP or APEX (`--fp16`) and FP16 half precision evaluation (`--fp16\_full\_eval`) can only be used on CUDA or NPU devices.

W0409 23:25:53.419351 3354519 site-packages/torch/distributed/elastic/multiprocessing/api.py:897] Sending process 3354522 closing signal SIGTERM

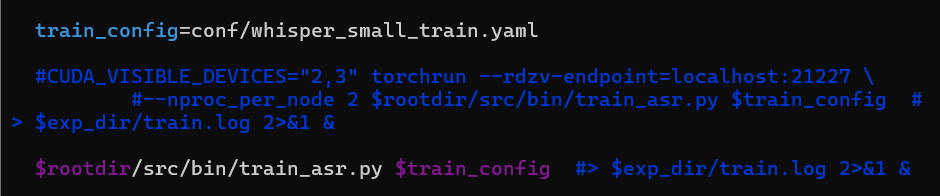
E0409 23:25:53.427443 3354519 site-packages/torch/distributed/elastic/multiprocessing/api.py:869] failed (exitcode: 1) local\_rank: 0 (pid: 3354521) of binary: /home/klp65/.conda/envs/spapl\_kidsasr/bin/python

Solution: set fp16 to False in yaml config file.

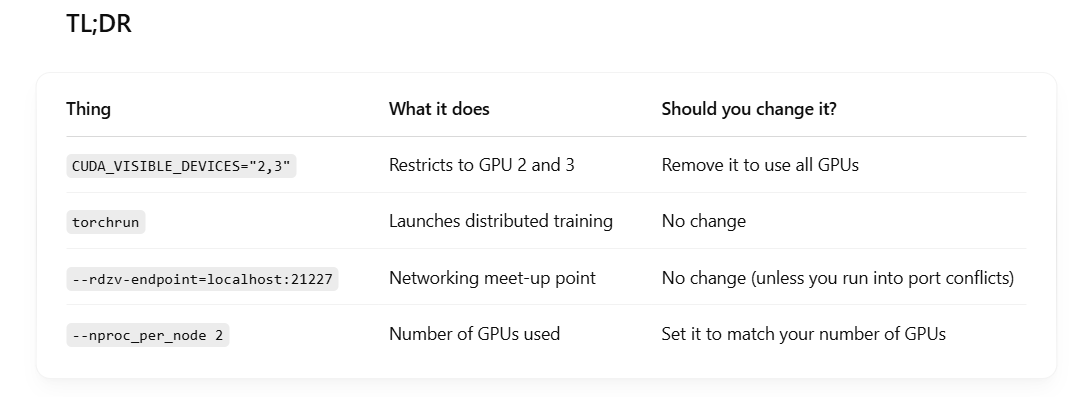
**Issue 2:**

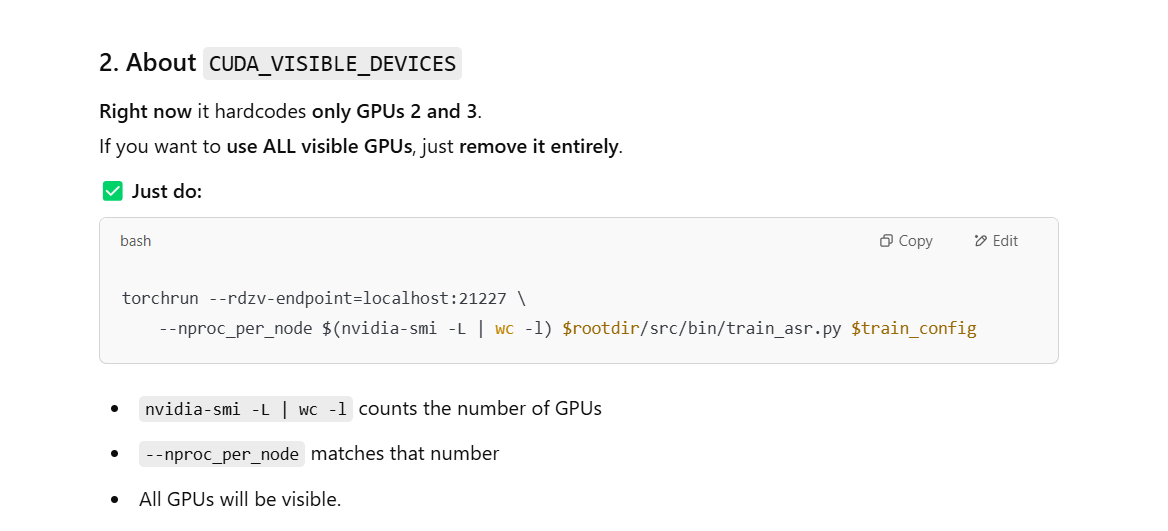
****

Remove line 71 in run\_whisper.sh since only 1 GPU, run .py directly



Trying to Adapt Code for Distributed Training (Multiple GPUs)

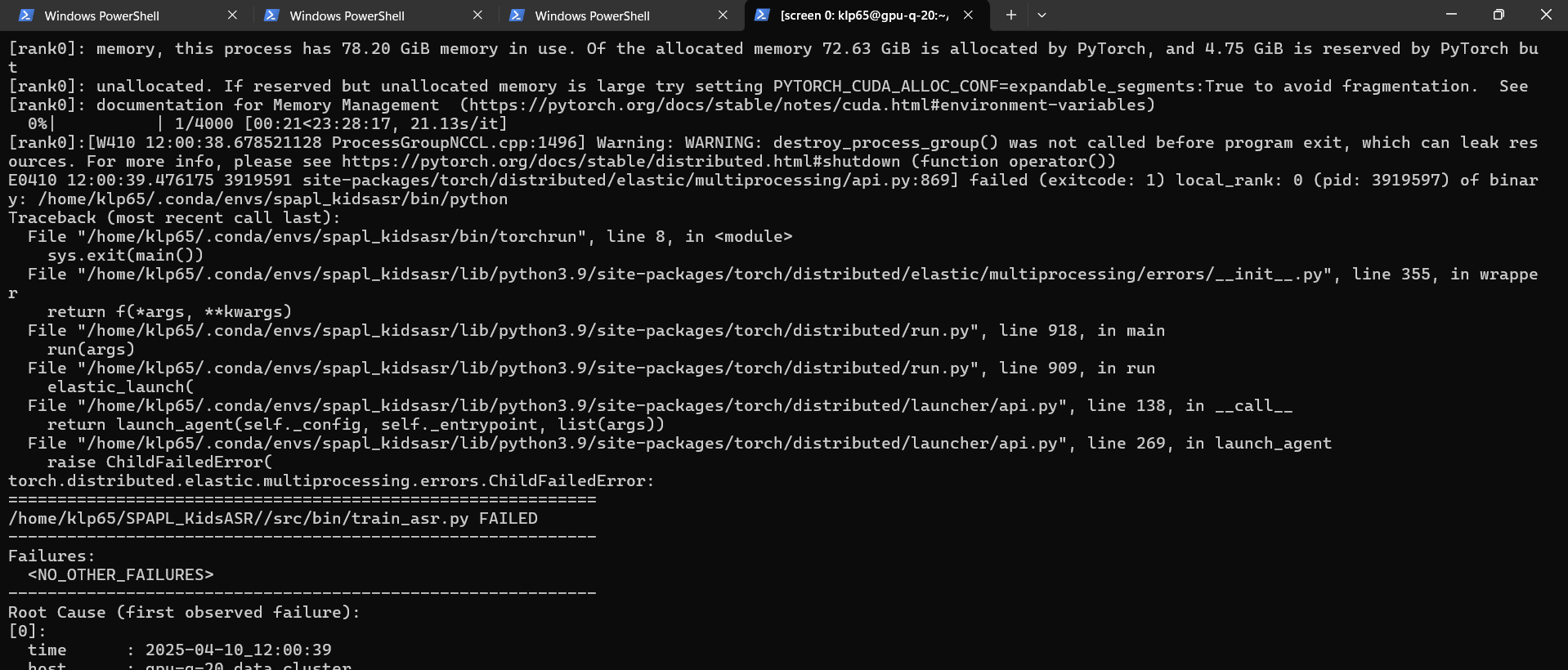




torchrun --rdzv-endpoint=localhost:21227 \

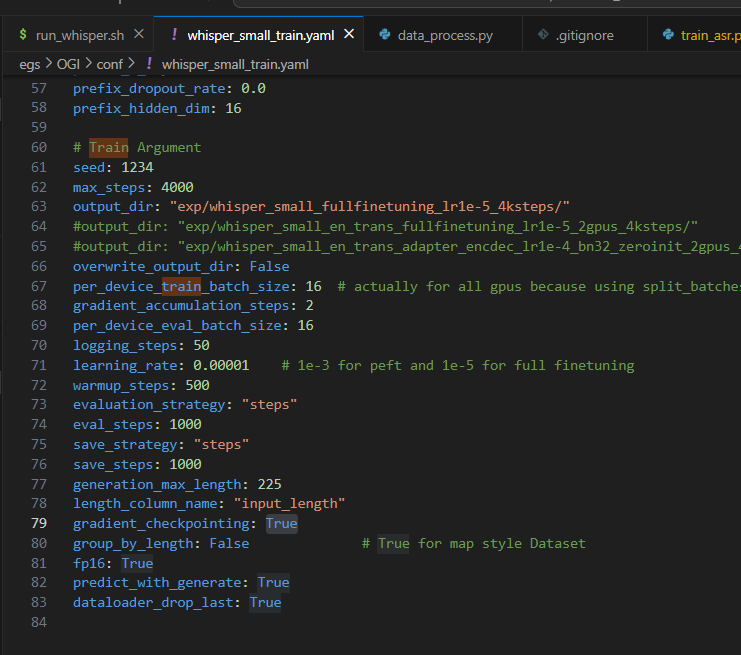
--nproc\_per\_node $(nvidia-smi -L | wc -l) $rootdir/src/bin/train\_asr.py $train\_config

But this causes out-of-memory error:

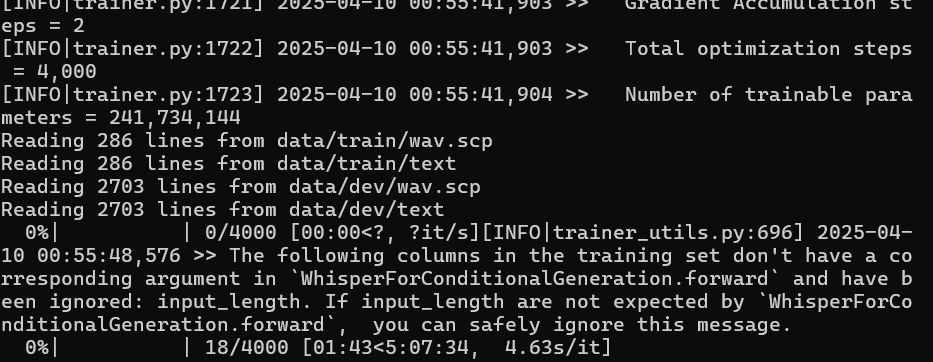


**Issue 3:**

AttributeError: 'WhisperForConditionalGeneration' object has no attribute '\_set\_gradient\_checkpointing'



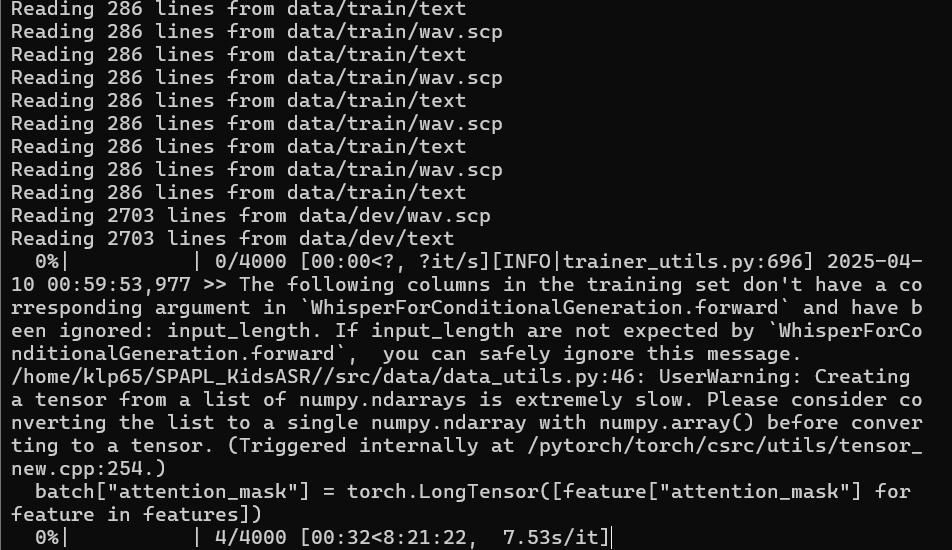
Gradient checkpointing changed to False



Training Success!

**Testing Stage 2 (All DA – not PIF, No PEFT):**

Works!

****

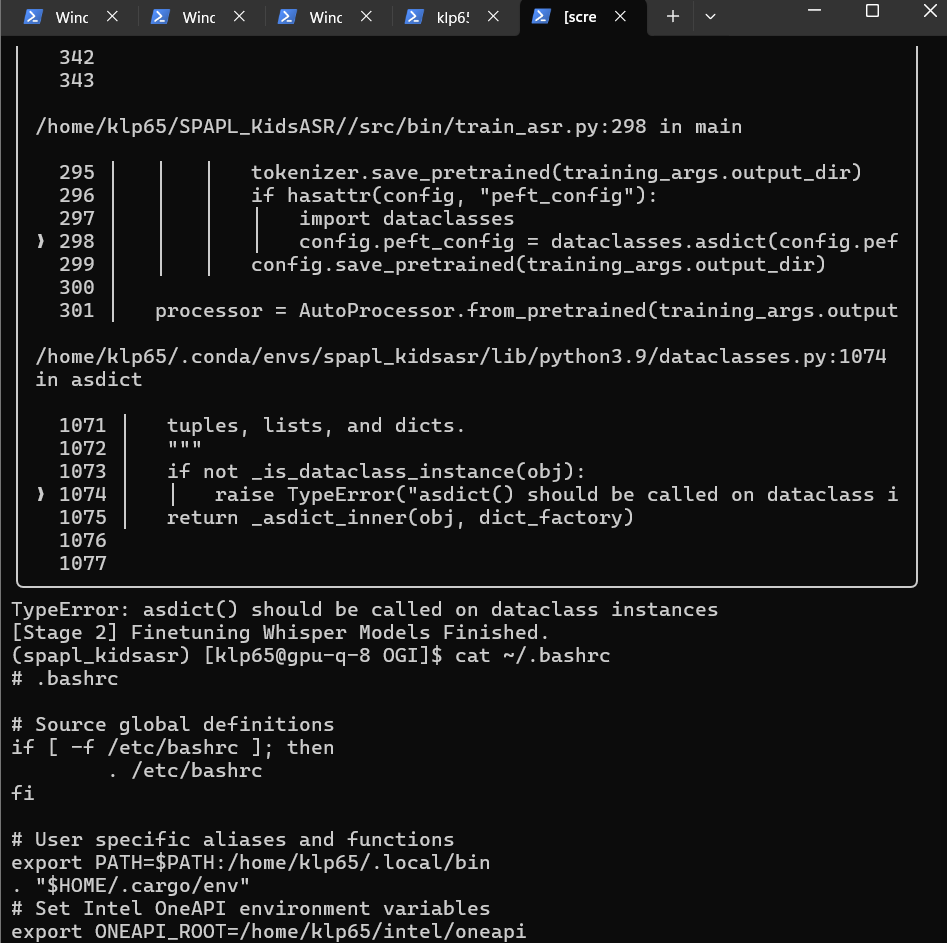
**Testing Stage 2 (All DA – PIF + PP, No PEFT):**

Error. May not want to explore this.

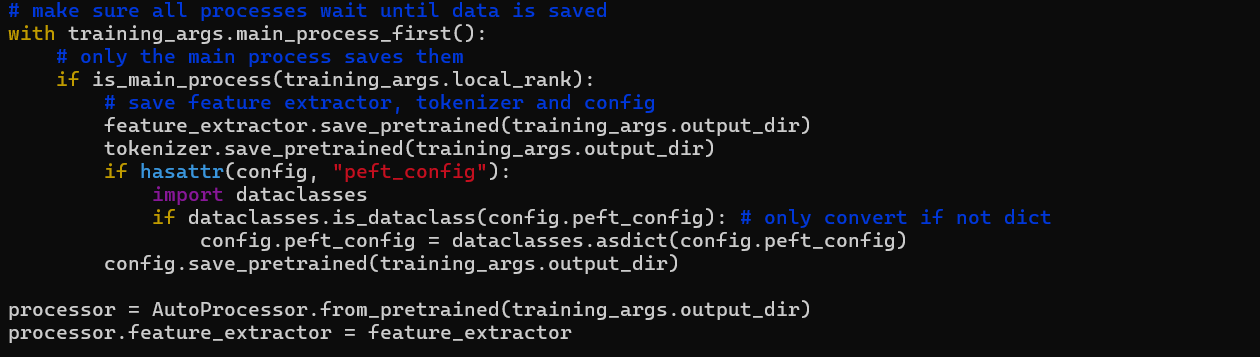
****

**Testing Stage 2 (No DA, lora adaptor)**

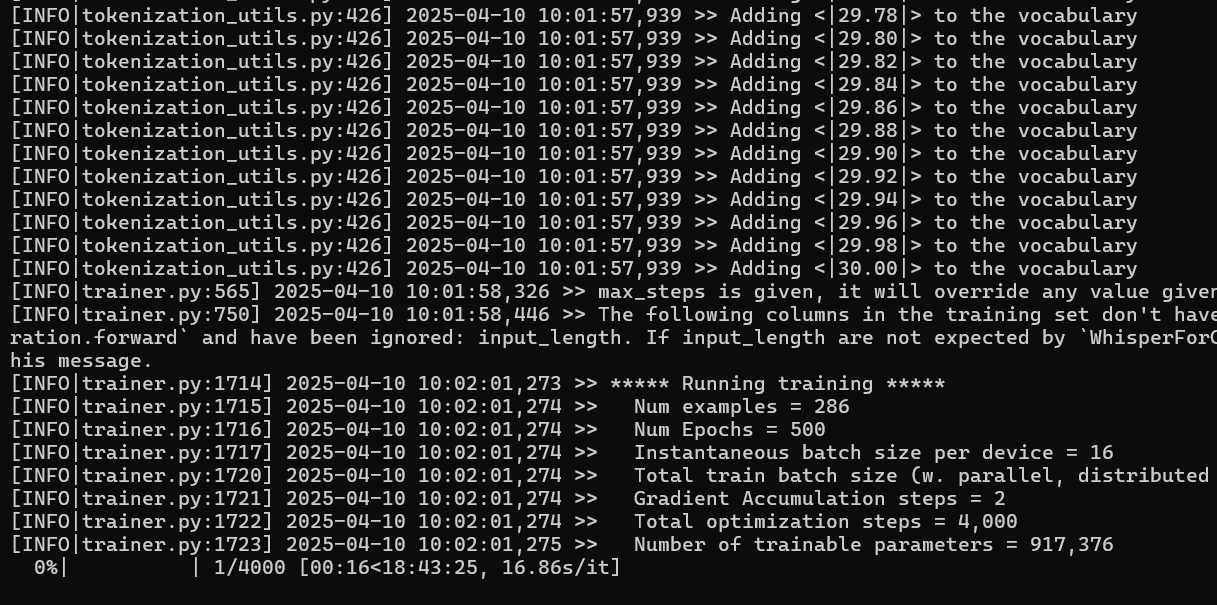
Error:



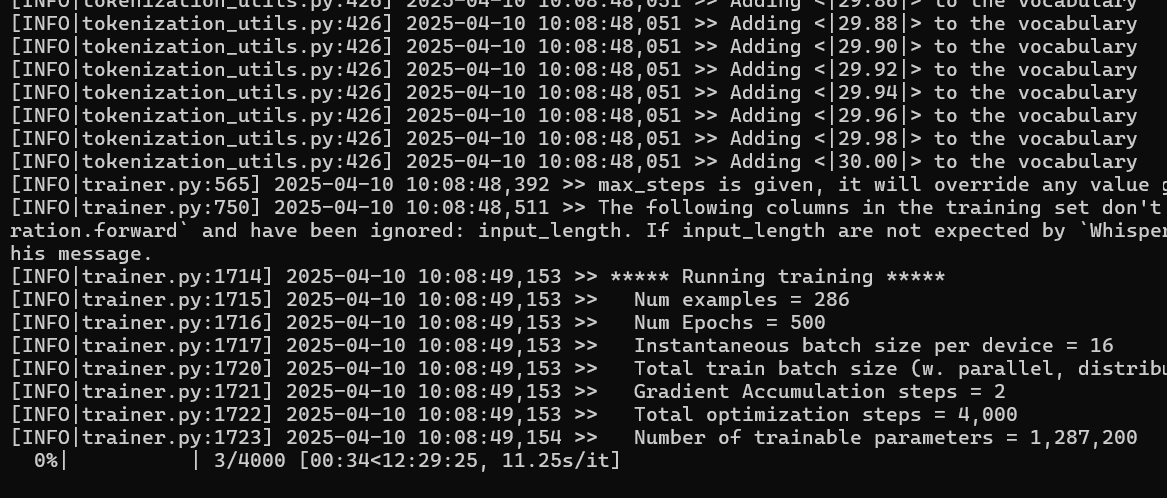
Resolve by:

****

Works!

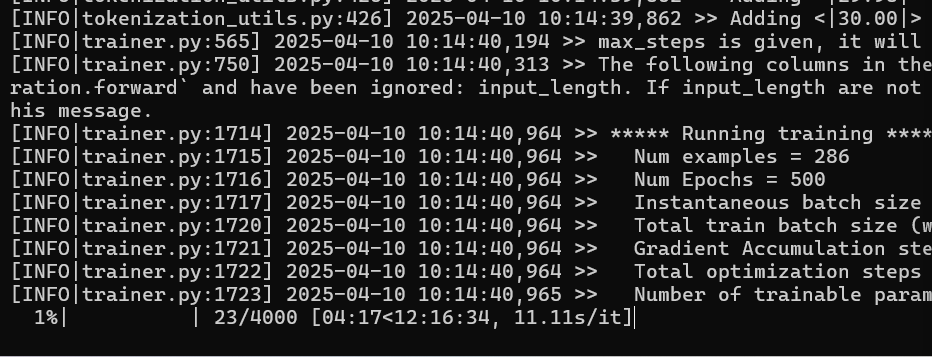
****

**Testing Stage 3 (No DA, adaptor)**

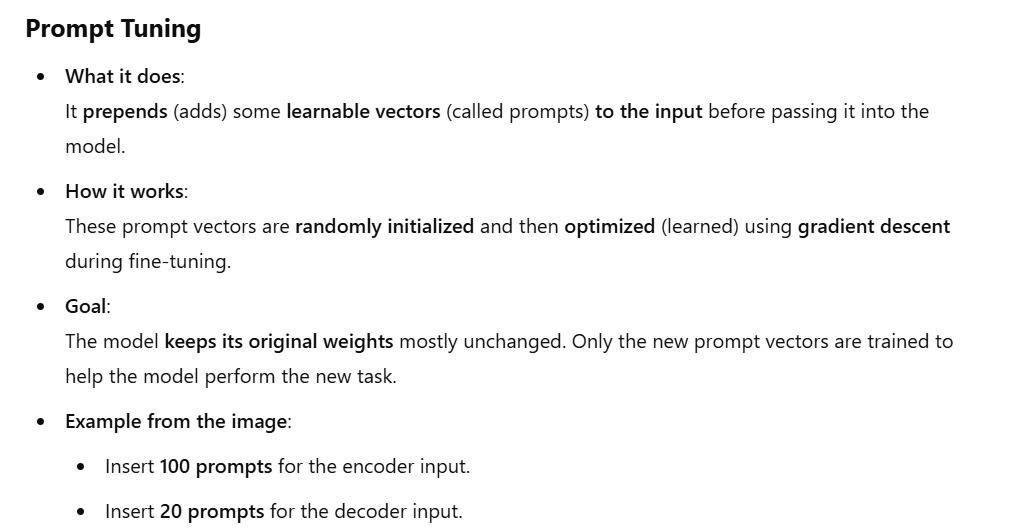
****

Works!

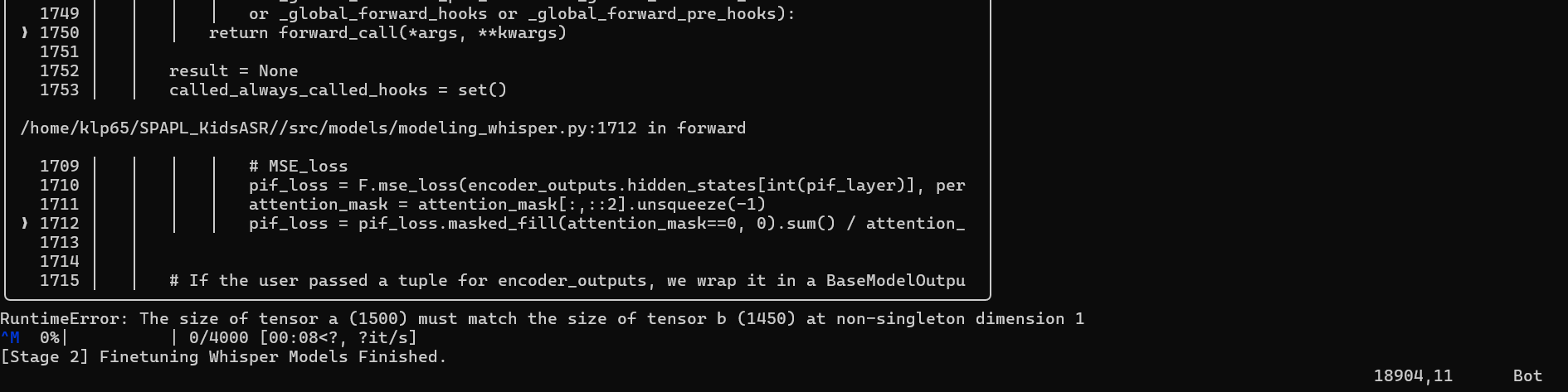
**Testing Stage 3 (No DA, prompt tuning)**

****

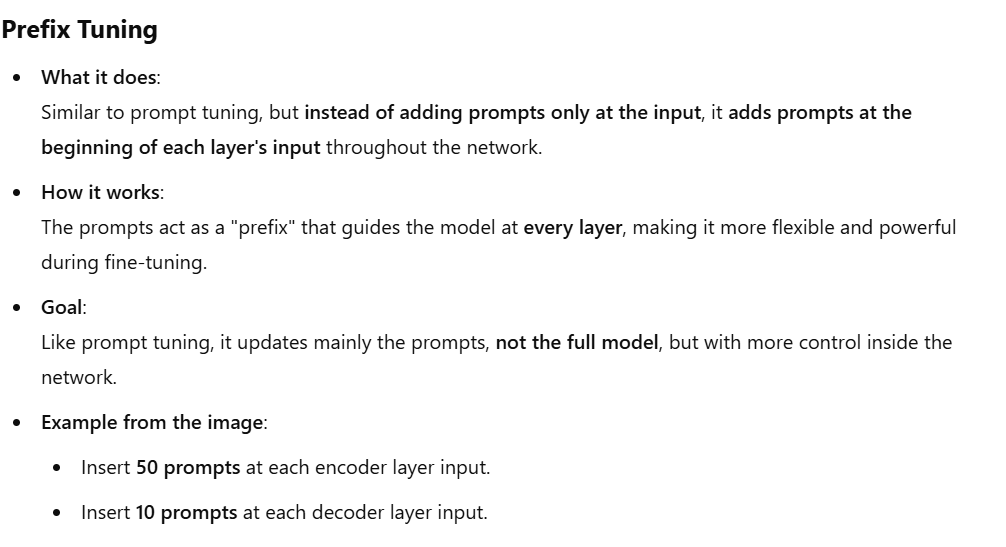
Works!



**Testing Stage 3 (No DA, prefix tuning)**

****

Error. (Read up on prefix tuning)



**Unsolved Error**

**Stage 3 Untested**

(same code as Stage 1 except that takes model from checkpoint)

**Testing MyST corpus code**

Train and development data:

    name: 'train'

    scp\_path: data/train\_filter/wav.scp

    text\_label: data/train\_filter/text

    name: 'development'

    scp\_path: data/select\_valid/wav.scp

text\_label: data/select\_valid/text

Test data:

data/test\_filter\_lt30

data/test\_filter\_gt30

data/development\_filter