Instructions for running Llama-2 on MIT SuperCloud

1. Set up SuperCloud account, follow SuperCloud docs to get SSH key set up.
2. Create a python script for downloading model you need, mine looked like this:

```python
from huggingface_hub import snapshot_download
from transformers import AutoModel, AutoTokenizer, AutoModelForCausalLM
import torch
import os
import yaml
import subprocess

# Main
if (__name__ == "__main__"):
    # Download model from the huggingface repo
    # The model is saved to disk where HF_HOME
    # HF_HOME is set to the local disk to enable filelocking

    with open('../src/config.yaml', "r") as f:
        config = yaml.load(f, Loader=yaml.FullLoader)
    hf_token = config["HUGGINGFACE_TOKEN"]
    subprocess.run(["huggingface-cli", "login", "--token", hf_token])
    repo_id="meta-llama/LlamaGuard-7b"
    tokenizer = AutoTokenizer.from_pretrained(repo_id)
    model = AutoModelForCausalLM.from_pretrained(repo_id)
```

   (Note: you need to get approved by Meta to use Llama-2, you can apply here: https://llama.meta.com/llama-downloads/, this may take a few days)
3. Create a bash script for running Python script to download HuggingFace model. Mine looked like this:

```bash
#!/bin/bash
#SBATCH --job-name=hf-download
#SBATCH -c 5
#SBATCH --partition=download
#SBATCH -o %j.log
source /etc/profile
module purge
module load anaconda/2023b

#Set HF_HOME environment variable to the local disk.
# The local disk is where filelocking is enabled.
HF_HOME=/state/partition1/user/$USER/huggingface
mkdir -p $HF_HOME
export HF_HOME
HF_LOCAL_DIR=$HOME/.cache/huggingfacei
mkdir -d $HF_LOCAL_DIR
# The model files will be saved to HF_HOME
python -u download.py

#copy the model from HF_HOME back to my  home directory
cp -rf $HF_HOME/* $HF_LOCAL_DIR
rm -rf $HF_HOME
```

   - Make sure you use download partition, this is the only partition that can connect to HuggingFace online to access the model
   - The directory $Home/.cache/huggingface (ignore the typo huggingfacei) is where we store the Llama-2 model tensors
   - Note: you may need to pip install libraries onto the module that you are using (I am using anaconda/2023b here) such as transformers, torch, etc.
4. Run bash script to download model
5. Import your script where you need to use Llama-2. I used mine in the following script called llama_script.py:

```python
from transformers import AutoTokenizer, AutoModelForCausalLM
import torch
import os
import yaml
import subprocess
import pandas as pd
import sys

def main():

#    model_id = "../models/llama/"
#    device = "cuda"
    model_path = sys.argv[1]
    print(model_path)
    dtype = torch.bfloat16
    tokenizer = AutoTokenizer.from_pretrained(model_path, local_files_only=True)
    model = AutoModelForCausalLM.from_pretrained(model_path, local_files_only=True)
    # tokenizer = AutoTokenizer.from_pretrained(model_id)
    # model = AutoModelForCausalLM.from_pretrained(model_id, torch_dtype=dtype, device_map=device)

    df = pd.read_csv('../data/pca_data.csv')

    request = list(df['prompt'])
    response = list(df['response'])
    labels = list(df['label'])
    print('1')
    def moderate_with_template(chat):
#        input_ids = tokenizer.apply_chat_template(chat, return_tensors="pt").to(device)
        input_ids = tokenizer.apply_chat_template(chat, return_tensors="pt").to("cpu")
        output = model.generate(input_ids=input_ids, max_new_tokens=100, pad_token_id=0)
        print('here!')
        prompt_len = input_ids.shape[-1]
        return tokenizer.decode(output[0][prompt_len:], skip_special_tokens=True)
```

(Note, you need to set local_files_only=True when importing; you can ignore the lines of code under this)

6. Create bash script to run your python script in which you use your Llama model. Mine looked like this:

```bash
#!/bin/bash
#SBATCH --job-name=hf-load
#SBATCH -o %j.log
#SBATCH --partition=xeon-g6-volta
#SBATCH --gres=gpu:volta:1
#SBATCH --cpus-per-task=20

#huggingface/hub/models--meta-llama--LlamaGuard-7b/snapshots/6da85dd9ca949a77ecdf4df13241b9b6ccd38a61
source /etc/profile
module purge
module load anaconda/2023b

HF_LOCAL_DIR=$HOME/.cache/huggingface
#HF_LOCAL_MODEL_DIR=$HF_LOCAL_DIR/hub/models--meta-llama--LlamaGuard-7b/snapshots/6da85dd9ca949a77ecdf4df13241b9b6ccd38a61
HF_LOCAL_MODEL_DIR=$HOME/.cache/huggingface/hub/models--meta-llama--LlamaGuard-7b/snapshots/6da85dd9ca949a77ecdf4df13241b9b6ccd38a61
export HF_HOME=$HF_LOCAL_DIR

python -u llama_script.py $HF_LOCAL_MODEL_DIR
```

- You will need to go to $HOME/.cache/huggingface/hub/models–meta-llama–LlamaGuard-7b/snapshots to see what your unique hash is, and paste it into this bash script
- Make sure you use the same partitions as shown here–this ensures you run your script on a GPU, which is the only way we can execute Llama-2