Installation and Load packages

```
!pip install datasets peft -qq
!pip install accelerate -qq
!pip install bitsandbytes -qq
!pip install trl -qq

pip show peft

Name: peft
Version: 0.5.0
Summary: Parameter-Efficient Fine-Tuning (PEFT)
Home-page: https://github.com/huggingface/peft
Author: The HuggingFace team
Author-email: sourab@huggingface.co
License: Apache
Location: /opt/conda/lib/python3.10/site-packages
Requires: accelerate, numpy, packaging, psutil, pyyaml, safetensors,
torch, tqdm, transformers
Required-by: auto-gptq
Note: you may need to restart the kernel to use updated packages.

!pip install torch==2.2.0 torchvision==0.17.0 torchaudio==2.2.0 --
index-url https://download.pytorch.org/whl/cu118
!pip install --upgrade --pre transformers accelerate --extra-index-url
https://download.pytorch.org/whl/cu118
!pip install bitsandbytes==0.43.2 --prefer-binary --extra-index-url
https://pypi.org/simple

Defaulting to user installation because normal site-packages is not
writeable
Looking in indexes: https://download.pytorch.org/whl/cu118
Collecting torch==2.2.0
  Downloading https://download.pytorch.org/whl/cu118/torch-
2.2.0%2Bcu118-cp310-cp310-linux_x86_64.whl (811.7 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 811.7/811.7 MB 1.6 MB/s eta
0:00:0000:0100:01
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.2/6.2 MB 54.6 MB/s eta
0:00:0000:0100:01
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.3/3.3 MB 66.6 MB/s eta
0:00:00:00:01
anylinux1_x86_64.whl (728.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 728.5/728.5 MB 1.8 MB/s eta
0:00:0000:0100:01
ent already satisfied: typing-extensions>=4.8.0 in
/home/student/.local/lib/python3.10/site-packages (from torch==2.2.0)
(4.10.0)
Requirement already satisfied: networkx in
```

```
/opt/conda/lib/python3.10/site-packages (from torch==2.2.0) (3.1)
Collecting nvidia-cuda-runtime-cu11==11.8.89
  Downloading
https://download.pytorch.org/whl/cu118/nvidia_cuda_runtime_cu11-
11.8.89-py3-none-manylinux1_x86_64.whl (875 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 875.6/875.6 kB 54.1 MB/s eta
0:00:00
anylinux1_x86_64.whl (13.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.1/13.1 MB 65.9 MB/s eta
0:00:0000:0100:01
anylinux1_x86_64.whl (417.9 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 417.9/417.9 MB 3.0 MB/s eta
0:00:0000:0100:01
ent already satisfied: sympy in /opt/conda/lib/python3.10/site-
packages (from torch==2.2.0) (1.12)
Collecting nvidia-cusolver-cu11==11.4.1.48
  Downloading
https://download.pytorch.org/whl/cu118/nvidia_cusolver_cu11-11.4.1.48-
py3-none-manylinux1_x86_64.whl (128.2 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 128.2/128.2 MB 10.4 MB/s eta
0:00:0000:0100:01
anylinux1_x86_64.whl (168.4 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 168.4/168.4 MB 8.0 MB/s eta
0:00:0000:0100:01
anylinux1_x86_64.whl (204.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 204.1/204.1 MB 6.5 MB/s eta
0:00:0000:0100:01
ent already satisfied: fsspec in
/home/student/.local/lib/python3.10/site-packages (from torch==2.2.0)
(2024.2.0)
Requirement already satisfied: filelock in
/home/student/.local/lib/python3.10/site-packages (from torch==2.2.0)
(3.13.1)
Collecting nvidia-nccl-cu11==2.19.3
  Downloading https://download.pytorch.org/whl/cu118/nvidia_nccl_cu11-
2.19.3-py3-none-manylinux1_x86_64.whl (135.3 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 135.3/135.3 MB 9.9 MB/s eta
0:00:0000:0100:01
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (167.9 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 167.9/167.9 MB 4.3 MB/s eta
0:00:0000:0100:01
anylinux1_x86_64.whl (58.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.1/58.1 MB 23.0 MB/s eta
0:00:0000:0100:01
anylinux1_x86_64.whl (99 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.1/99.1 kB 15.5 MB/s eta
0:00:00
anylinux1_x86_64.whl (23.2 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23.2/23.2 MB 57.5 MB/s eta
```

```
0:00:0000:0100:01
ent already satisfied: jinja2 in
/home/student/.local/lib/python3.10/site-packages (from torch==2.2.0)
(3.1.3)
Requirement already satisfied: numpy in
/home/student/.local/lib/python3.10/site-packages (from
torchvision==0.17.0) (1.26.4)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/home/student/.local/lib/python3.10/site-packages (from
torchvision==0.17.0) (10.2.0)
Requirement already satisfied: requests in
/home/student/.local/lib/python3.10/site-packages (from
torchvision==0.17.0) (2.32.3)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/student/.local/lib/python3.10/site-packages (from jinja2-
>torch==2.2.0) (2.1.5)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/student/.local/lib/python3.10/site-packages (from requests-
>torchvision==0.17.0) (2.2.1)
Requirement already satisfied: certifi>=2017.4.17 in
/home/student/.local/lib/python3.10/site-packages (from requests-
>torchvision==0.17.0) (2024.2.2)
Requirement already satisfied: idna<4,>=2.5 in
/home/student/.local/lib/python3.10/site-packages (from requests-
>torchvision==0.17.0) (3.6)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/student/.local/lib/python3.10/site-packages (from requests-
>torchvision==0.17.0) (3.3.2)
Requirement already satisfied: mpmath>=0.19 in
/opt/conda/lib/python3.10/site-packages (from sympy->torch==2.2.0)
(1.3.0)
Installing collected packages: triton, nvidia-nvtx-cu11, nvidia-nccl-
cu11, nvidia-cusparse-cu11, nvidia-curand-cu11, nvidia-cufft-cu11,
nvidia-cuda-runtime-cu11, nvidia-cuda-nvrtc-cu11, nvidia-cuda-cupti-
cu11, nvidia-cublas-cu11, nvidia-cusolver-cu11, nvidia-cudnn-cu11,
torch, torchvision, torchaudio
  WARNING: The scripts convert-caffe2-to-onnx, convert-onnx-to-caffe2
and torchrun are installed in '/home/student/.local/bin' which is not
on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress
this warning, use --no-warn-script-location.
Successfully installed nvidia-cublas-cu11-11.11.3.6 nvidia-cuda-cupti-
cu11-11.8.87 nvidia-cuda-nvrtc-cu11-11.8.89 nvidia-cuda-runtime-cu11-
11.8.89 nvidia-cudnn-cu11-8.7.0.84 nvidia-cufft-cu11-10.9.0.58 nvidia-
curand-cu11-10.3.0.86 nvidia-cusolver-cu11-11.4.1.48 nvidia-cusparse-
cu11-11.7.5.86 nvidia-nccl-cu11-2.19.3 nvidia-nvtx-cu11-11.8.86 torch-
2.2.0+cu118 torchaudio-2.2.0+cu118 torchvision-0.17.0+cu118 triton-
2.2.0
Defaulting to user installation because normal site-packages is not
```

```
writeable
Looking in indexes: https://pypi.org/simple,
https://download.pytorch.org/whl/cu118
Requirement already satisfied: transformers in
/home/student/.local/lib/python3.10/site-packages (4.51.3)
Requirement already satisfied: accelerate in
/home/student/.local/lib/python3.10/site-packages (1.6.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in
/home/student/.local/lib/python3.10/site-packages (from transformers)
(0.30.2)
Requirement already satisfied: pyyaml>=5.1 in
/home/student/.local/lib/python3.10/site-packages (from transformers)
(6.0.1)
Requirement already satisfied: tokenizers<0.22,>=0.21 in
/home/student/.local/lib/python3.10/site-packages (from transformers)
(0.21.1)
Requirement already satisfied: safetensors>=0.4.3 in
/home/student/.local/lib/python3.10/site-packages (from transformers)
(0.5.3)

Requirement already satisfied: tqdm>=4.27 in
/home/student/.local/lib/python3.10/site-packages (from transformers)
(4.67.1)
Requirement already satisfied: regex!=2019.12.17 in
/opt/conda/lib/python3.10/site-packages (from transformers)
(2023.12.25)
Requirement already satisfied: filelock in
/home/student/.local/lib/python3.10/site-packages (from transformers)
(3.13.1)
Requirement already satisfied: requests in
/home/student/.local/lib/python3.10/site-packages (from transformers)
(2.32.3)
Requirement already satisfied: packaging>=20.0 in
/home/student/.local/lib/python3.10/site-packages (from transformers)
(24.0)
Requirement already satisfied: numpy>=1.17 in
/home/student/.local/lib/python3.10/site-packages (from transformers)
(1.26.4)
Requirement already satisfied: psutil in
/opt/conda/lib/python3.10/site-packages (from accelerate) (5.9.0)
Requirement already satisfied: torch>=2.0.0 in
/home/student/.local/lib/python3.10/site-packages (from accelerate)
(2.2.0+cu118)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/home/student/.local/lib/python3.10/site-packages (from huggingface-
hub<1.0,>=0.30.0->transformers) (4.10.0)
Requirement already satisfied: fsspec>=2023.5.0 in
/home/student/.local/lib/python3.10/site-packages (from huggingface-
hub<1.0,>=0.30.0->transformers) (2024.2.0)
Requirement already satisfied: nvidia-cublas-cu11==11.11.3.6 in
```

```
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (11.11.3.6)
Requirement already satisfied: nvidia-cusolver-cu11==11.4.1.48 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (11.4.1.48)
Requirement already satisfied: nvidia-nvtx-cu11==11.8.86 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (11.8.86)
Requirement already satisfied: triton==2.2.0 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (2.2.0)
Requirement already satisfied: nvidia-cusparse-cu11==11.7.5.86 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (11.7.5.86)
Requirement already satisfied: nvidia-cuda-runtime-cu11==11.8.89 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (11.8.89)
Requirement already satisfied: networkx in
/opt/conda/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (3.1)
Requirement already satisfied: nvidia-cufft-cu11==10.9.0.58 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (10.9.0.58)
Requirement already satisfied: jinja2 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (3.1.3)
Requirement already satisfied: nvidia-cuda-cupti-cu11==11.8.87 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (11.8.87)
Requirement already satisfied: nvidia-cudnn-cu11==8.7.0.84 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (8.7.0.84)
Requirement already satisfied: nvidia-curand-cu11==10.3.0.86 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (10.3.0.86)
Requirement already satisfied: sympy in
/opt/conda/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (1.12)
Requirement already satisfied: nvidia-cuda-nvrtc-cu11==11.8.89 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (11.8.89)
Requirement already satisfied: nvidia-nccl-cu11==2.19.3 in
/home/student/.local/lib/python3.10/site-packages (from torch>=2.0.0-
>accelerate) (2.19.3)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/student/.local/lib/python3.10/site-packages (from requests-
>transformers) (2.2.1)
Requirement already satisfied: idna<4,>=2.5 in
/home/student/.local/lib/python3.10/site-packages (from requests-
```

```
>transformers) (3.6)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/student/.local/lib/python3.10/site-packages (from requests-
>transformers) (3.3.2)
Requirement already satisfied: certifi>=2017.4.17 in
/home/student/.local/lib/python3.10/site-packages (from requests-
>transformers) (2024.2.2)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/student/.local/lib/python3.10/site-packages (from jinja2-
>torch>=2.0.0->accelerate) (2.1.5)
Requirement already satisfied: mpmath>=0.19 in
/opt/conda/lib/python3.10/site-packages (from sympy->torch>=2.0.0-
>accelerate) (1.3.0)
Defaulting to user installation because normal site-packages is not
writeable
Looking in indexes: https://pypi.org/simple, https://pypi.org/simple
Collecting bitsandbytes==0.43.2
  Downloading bitsandbytes-0.43.2-py3-none-manylinux_2_24_x86_64.whl
(137.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 137.5/137.5 MB 9.5 MB/s eta
0:00:0000:0100:01
ent already satisfied: torch in
/home/student/.local/lib/python3.10/site-packages (from
bitsandbytes==0.43.2) (2.2.0+cu118)
Requirement already satisfied: numpy in
/home/student/.local/lib/python3.10/site-packages (from
bitsandbytes==0.43.2) (1.26.4)
Requirement already satisfied: nvidia-cudnn-cu11==8.7.0.84 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (8.7.0.84)
Requirement already satisfied: networkx in
/opt/conda/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (3.1)
Requirement already satisfied: nvidia-cufft-cu11==10.9.0.58 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (10.9.0.58)
Requirement already satisfied: nvidia-cusparse-cu11==11.7.5.86 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (11.7.5.86)
Requirement already satisfied: typing-extensions>=4.8.0 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (4.10.0)
Requirement already satisfied: nvidia-nccl-cu11==2.19.3 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (2.19.3)
Requirement already satisfied: nvidia-nvtx-cu11==11.8.86 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (11.8.86)
Requirement already satisfied: filelock in
```

```
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (3.13.1)
Requirement already satisfied: jinja2 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (3.1.3)
Requirement already satisfied: nvidia-cusolver-cu11==11.4.1.48 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (11.4.1.48)
Requirement already satisfied: nvidia-cuda-cupti-cu11==11.8.87 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (11.8.87)
Requirement already satisfied: nvidia-cublas-cu11==11.11.3.6 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (11.11.3.6)
Requirement already satisfied: nvidia-curand-cu11==10.3.0.86 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (10.3.0.86)
Requirement already satisfied: nvidia-cuda-nvrtc-cu11==11.8.89 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (11.8.89)
Requirement already satisfied: sympy in
/opt/conda/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (1.12)
Requirement already satisfied: nvidia-cuda-runtime-cu11==11.8.89 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (11.8.89)
Requirement already satisfied: fsspec in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (2024.2.0)
Requirement already satisfied: triton==2.2.0 in
/home/student/.local/lib/python3.10/site-packages (from torch-
>bitsandbytes==0.43.2) (2.2.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/student/.local/lib/python3.10/site-packages (from jinja2->torch-
>bitsandbytes==0.43.2) (2.1.5)
Requirement already satisfied: mpmath>=0.19 in
/opt/conda/lib/python3.10/site-packages (from sympy->torch-
>bitsandbytes==0.43.2) (1.3.0)
Installing collected packages: bitsandbytes
Successfully installed bitsandbytes-0.43.2

!pip install wandb scikit-learn

Defaulting to user installation because normal site-packages is not
writeable
Collecting wandb
  Downloading wandb-0.19.10-py3-none-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (21.3 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.3/21.3 MB 45.7 MB/s eta
0:00:0000:0100:01
```

```
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.5/13.5 MB 63.4 MB/s eta
0:00:0000:0100:01
ent already satisfied: pyyaml in
/home/student/.local/lib/python3.10/site-packages (from wandb) (6.0.1)
Requirement already satisfied: platformdirs in
/opt/conda/lib/python3.10/site-packages (from wandb) (4.2.0)
Requirement already satisfied: pydantic<3 in
/home/student/.local/lib/python3.10/site-packages (from wandb) (2.6.4)
Collecting sentry-sdk>=2.0.0
  Downloading sentry_sdk-2.27.0-py2.py3-none-any.whl (340 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 340.8/340.8 kB 40.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.6/207.6 kB 31.5 MB/s eta
0:00:00
ent already satisfied: requests<3,>=2.0.0 in
/home/student/.local/lib/python3.10/site-packages (from wandb)
(2.32.3)
Requirement already satisfied: psutil>=5.0.0 in
/opt/conda/lib/python3.10/site-packages (from wandb) (5.9.0)
Requirement already satisfied: setuptools in
/opt/conda/lib/python3.10/site-packages (from wandb) (65.6.3)
Requirement already satisfied: typing-extensions<5,>=4.4 in
/home/student/.local/lib/python3.10/site-packages (from wandb)
(4.10.0)
Collecting setproctitle
  Downloading setproctitle-1.3.5-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux
2014_x86_64.whl (30 kB)
Requirement already satisfied: click!=8.0.0,>=7.1 in
/home/student/.local/lib/python3.10/site-packages (from wandb) (8.1.7)
Requirement already satisfied: protobuf!=4.21.0,!=5.28.0,<7,>=3.19.0
in /opt/conda/lib/python3.10/site-packages (from wandb) (4.25.3)
Collecting threadpoolctl>=3.1.0
  Downloading threadpoolctl-3.6.0-py3-none-any.whl (18 kB)
Requirement already satisfied: scipy>=1.6.0 in
/opt/conda/lib/python3.10/site-packages (from scikit-learn) (1.11.2)
Collecting joblib>=1.2.0
  Downloading joblib-1.4.2-py3-none-any.whl (301 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 301.8/301.8 kB 36.2 MB/s eta
0:00:00
ent already satisfied: numpy>=1.19.5 in
/home/student/.local/lib/python3.10/site-packages (from scikit-learn)
(1.26.4)
Requirement already satisfied: six>=1.4.0 in
/home/student/.local/lib/python3.10/site-packages (from docker-
pycreds>=0.4.0->wandb) (1.16.0)
Collecting gitdb<5,>=4.0.1
  Downloading gitdb-4.0.12-py3-none-any.whl (62 kB)
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.8/62.8 kB 12.3 MB/s eta
0:00:00
ent already satisfied: annotated-types>=0.4.0 in
/home/student/.local/lib/python3.10/site-packages (from pydantic<3-
>wandb) (0.6.0)
Requirement already satisfied: pydantic-core==2.16.3 in
/home/student/.local/lib/python3.10/site-packages (from pydantic<3-
>wandb) (2.16.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/student/.local/lib/python3.10/site-packages (from
requests<3,>=2.0.0->wandb) (3.3.2)
Requirement already satisfied: certifi>=2017.4.17 in
/home/student/.local/lib/python3.10/site-packages (from
requests<3,>=2.0.0->wandb) (2024.2.2)
Requirement already satisfied: idna<4,>=2.5 in
/home/student/.local/lib/python3.10/site-packages (from
requests<3,>=2.0.0->wandb) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/student/.local/lib/python3.10/site-packages (from
requests<3,>=2.0.0->wandb) (2.2.1)
Collecting smmap<6,>=3.0.1
  Downloading smmap-5.0.2-py3-none-any.whl (24 kB)
Installing collected packages: threadpoolctl, smmap, setproctitle,
sentry-sdk, joblib, docker-pycreds, scikit-learn, gitdb, gitpython,
wandb
  WARNING: The scripts wandb and wb are installed in
'/home/student/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress
this warning, use --no-warn-script-location.
Successfully installed docker-pycreds-0.4.0 gitdb-4.0.12 gitpython-
3.1.44 joblib-1.4.2 scikit-learn-1.6.1 sentry-sdk-2.27.0 setproctitle-
1.3.5 smmap-5.0.2 threadpoolctl-3.6.0 wandb-0.19.10
```

```
!pip show transformers
```

```
Name: transformers
Version: 4.51.3
Summary: State-of-the-art Machine Learning for JAX, PyTorch and
TensorFlow
Home-page: https://github.com/huggingface/transformers
Author: The Hugging Face team (past and future) with the help of all
our contributors
(https://github.com/huggingface/transformers/graphs/contributors)
Author-email: transformers@huggingface.co
License: Apache 2.0 License
Location: /home/student/.local/lib/python3.10/site-packages
Requires: filelock, huggingface-hub, numpy, packaging, pyyaml, regex,
requests, safetensors, tokenizers, tqdm
Required-by: auto-gptq, optimum, peft, trl
```

```
import os

# Disable tokenizer parallelism to avoid the warning
os.environ["TOKENIZERS_PARALLELISM"] = "false"

import peft
print(peft.__version__)
print(peft.__file__)

import sys
for path in sys.path:
    print(path)
```

```
0.5.0
/opt/conda/lib/python3.10/site-packages/peft/__init__.py
/opt/conda/lib/python310.zip
/opt/conda/lib/python3.10
/opt/conda/lib/python3.10/lib-dynload

/home/student/.local/lib/python3.10/site-packages
/opt/conda/lib/python3.10/site-packages
/opt/conda/lib/python3.10/site-packages/mpmath-1.2.1-py3.10.egg
```

```
# !pip uninstall peft
```

GPU - details

```
import torch

print("Torch version:", torch.__version__)
print("CUDA available:", torch.cuda.is_available())

if torch.cuda.is_available():
    print("Device name:", torch.cuda.get_device_name(0))
else:
    print("No GPU detected.")
```

```
Torch version: 2.2.0+cu118
CUDA available: True
Device name: Tesla T4
```

Load libraries, Login HuggingFace API & WandB API

- **HuggingFace API:** To get access of Model Llama-3.2 (3 Billion)
- **WandB (Weigths & Biases):** To supervise perform of model and hyperparameter Tuning

```
# from google.colab import userdata
from huggingface_hub import login

login(token="YOUR_HF_API_KEY")
# Access Key for llama Model (HuggingFace)

from datasets import load_dataset, Dataset
from sklearn.model_selection import train_test_split

from transformers import (
    AutoTokenizer,
    AutoModelForCausalLM,
    TrainingArguments,
    DataCollatorForLanguageModeling,
    Trainer,
    BitsAndBytesConfig,
    HfArgumentParser,
    pipeline,
    logging,
    EarlyStoppingCallback
)

from transformers.trainer_callback import TrainerCallback,
TrainerState, TrainerControl

from peft import (
    LoraConfig,
    PeftModel,
    prepare_model_for_kbit_training,
    get_peft_model,
)


from bitsandbytes.optim import AdamW8bit
import os, torch, wandb
from trl import SFTTrainer, setup_chat_format
```

WandB - For plot Training

```
# for hyperparameter tuning report
wandb.login()
# YOUR_WANDB_API_KEY
```

```
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server
locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here:
https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter:
```

```
 ........

wandb: WARNING If you're specifying your api key in code, ensure this
code is not shared publicly.
wandb: WARNING Consider setting the WANDB_API_KEY environment
variable, or running `wandb login` from the command line.
wandb: No netrc file found, creating one.
wandb: Appending key for api.wandb.ai to your netrc file:
/home/student/.netrc
wandb: Currently logged in as: yashnayi00 (yashnayi00-university-of-
new-haven) to https://api.wandb.ai. Use `wandb login --relogin` to
force relogin

True
```

Load Llama-3.2-3B model

```
model_name = "meta-llama/Llama-3.2-3B"

bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16,
    bnb_4bit_use_double_quant=False
)


tokenizer = AutoTokenizer.from_pretrained(model_name)

base_model = AutoModelForCausalLM.from_pretrained(
    model_name,
    device_map="auto",
    quantization_config=bnb_config,
    attn_implementation="eager",
)

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

tokenizer.padding_side = "right"

base_model.config.pretraining_tp = 1
base_model.config.use_cache = False

{"model_id":"3361be8c80674f96bb50166eab915fa5","version_major":2,"vers
ion_minor":0}

{"model_id":"552d0c0e807e4821907602766ad452a3","version_major":2,"vers
ion_minor":0}
```

{"model_id":"962270397da94f87a0c668988608403b","version_major":2,"version_minor":0}

{"model_id":"d21184089d574183859584c512c84e58","version_major":2,"version_minor":0}

{"model_id":"52711e8ee2bd493baf4a21dfdbfc8619","version_major":2,"version_minor":0}

{"model_id":"613e9b90e0c14a6a93ee6785306a7109","version_major":2,"version_minor":0}

{"model_id":"600a1f91718140b29634f396fdb8231b","version_major":2,"version_minor":0}

{"model_id":"c3da2516242f4e65a5f4275ac603125a","version_major":2,"version_minor":0}

{"model_id":"7c4c4da0beee41e2b408dc8fde723cc6","version_major":2,"version_minor":0}

{"model_id":"629f43eea07b4b7a9e4acf701093d241","version_major":2,"version_minor":0}

```
print(f"meta-llama/Llama-3.2-3B:\n\n{base_model}")
```

```
meta-llama/Llama-3.2-3B:

LlamaForCausalLM(
  (model): LlamaModel(
    (embed_tokens): Embedding(128256, 3072)
    (layers): ModuleList(
      (0-27): 28 x LlamaDecoderLayer(
        (self_attn): LlamaAttention(
          (q_proj): Linear4bit(in_features=3072, out_features=3072,
bias=False)
          (k_proj): Linear4bit(in_features=3072, out_features=1024,
bias=False)
          (v_proj): Linear4bit(in_features=3072, out_features=1024,
bias=False)
          (o_proj): Linear4bit(in_features=3072, out_features=3072,
bias=False)
        )
        (mlp): LlamaMLP(
          (gate_proj): Linear4bit(in_features=3072, out_features=8192,
bias=False)
          (up_proj): Linear4bit(in_features=3072, out_features=8192,
bias=False)
          (down_proj): Linear4bit(in_features=8192, out_features=3072,
bias=False)
          (act_fn): SiLU()
```

```
        )
        (input_layernorm): LlamaRMSNorm((3072,), eps=1e-05)
        (post_attention_layernorm): LlamaRMSNorm((3072,), eps=1e-05)
      )
    )
    (norm): LlamaRMSNorm((3072,), eps=1e-05)
    (rotary_emb): LlamaRotaryEmbedding()
  )
  (lm_head): Linear(in_features=3072, out_features=128256, bias=False)
)

print(f"{base_model.config}")

LlamaConfig {
  "_attn_implementation_autoset": true,
  "architectures": [
    "LlamaForCausalLM"
  ],
  "attention_bias": false,
  "attention_dropout": 0.0,
  "bos_token_id": 128000,
  "eos_token_id": 128001,
  "head_dim": 128,
  "hidden_act": "silu",
  "hidden_size": 3072,
  "initializer_range": 0.02,
  "intermediate_size": 8192,
  "max_position_embeddings": 131072,
  "mlp_bias": false,
  "model_type": "llama",
  "num_attention_heads": 24,
  "num_hidden_layers": 28,
  "num_key_value_heads": 8,
  "pretraining_tp": 1,
  "quantization_config": {
    "_load_in_4bit": true,
    "_load_in_8bit": false,
    "bnb_4bit_compute_dtype": "bfloat16",
    "bnb_4bit_quant_storage": "uint8",
    "bnb_4bit_quant_type": "nf4",
    "bnb_4bit_use_double_quant": false,
    "llm_int8_enable_fp32_cpu_offload": false,
    "llm_int8_has_fp16_weight": false,
    "llm_int8_skip_modules": null,
    "llm_int8_threshold": 6.0,
    "load_in_4bit": true,
    "load_in_8bit": false,
    "quant_method": "bitsandbytes"
  },
  "rms_norm_eps": 1e-05,
```

```
  "rope_scaling": {
    "factor": 32.0,
    "high_freq_factor": 4.0,
    "low_freq_factor": 1.0,
    "original_max_position_embeddings": 8192,
    "rope_type": "llama3"
  },
  "rope_theta": 500000.0,
  "tie_word_embeddings": true,
  "torch_dtype": "float16",
  "transformers_version": "4.51.3",
  "use_cache": false,
  "vocab_size": 128256
}
```

Trainable parameters - Model

```
def trainable_parameters(model):
    """
    Prints the number of trainable parameters in the model.
    """
    trainable_params = 0
    all_param = 0
    for _, param in model.named_parameters():
        all_param += param.numel()
        if param.requires_grad:
            trainable_params += param.numel()
    return f"- Trainable model parameters: {trainable_params}.\n- All model parameters: {all_param}.\n- Percentage of trainable model parameters: {100 * trainable_params / all_param:.2f}%"

print(trainable_parameters(base_model))
```

```
- Trainable model parameters: 394177536.
- All model parameters: 1803463680.
- Percentage of trainable model parameters: 21.86%
```

Assign datasetPH.json

Data is split in to train and test.

- Train size: 80%
- Test size: 20%

```
# import json
# with open("./dataset/policy_training_data.jsonl", "r") as f:
#     data = json.load(f)

# if isinstance(data, dict):
```

```
#     print("Data is a dictionary. Converting values to a list for splitting.")
#     data = list(data.values())

# train_data, test_data = train_test_split(data, test_size=0.2, random_state=42)

# with open("./dataset/trainset/rp_train_datasetPH.json", "w") as f:
#     json.dump(train_data, f, indent=2)

# with open("./dataset/testset/rp_test_datasetPH.json", "w") as f:
#     json.dump(test_data, f, indent=2)

# print(f"Train size: {len(train_data)}")
# print(f"Test size: {len(test_data)}")

data = load_dataset("json", data_files="dataset/policy_data.jsonl")
data
```

```
DatasetDict({
    train: Dataset({
        features: ['instruction', 'response'],
        num_rows: 1215
    })
})
```

```
split_data = data["train"].train_test_split(test_size=0.2, seed=42)

print(split_data)
```

```
DatasetDict({
    train: Dataset({
        features: ['instruction', 'response'],
        num_rows: 972
    })
    test: Dataset({
        features: ['instruction', 'response'],
        num_rows: 243
    })
})
```

```
split_data['train'][0]
```

```
{'instruction': 'How does unemployment influence policies on sexual health education in the USA?',
 'response': 'Unemployment above 6 percent increases sexual health education policies by 10-15 percent. States target idle populations to cut STDs, but racial disparities and lack of insurance reduce reach by 10-15 percent.'}
```

Tokenization of dataset and normalization

```
# def tokenize_function(examples):
#     texts = []
#     for i in range(len(examples["title"])):
#         entry_parts = []

#         for key in examples.keys():
#             value = examples[key][i]
#             if isinstance(value, dict):
#                 for subkey, subval in value.items():
#                     entry_parts.append(f"{key}.{subkey}: {subval}")
#             elif isinstance(value, list):
#                 entry_parts.append(f"{key}: {', '.join(map(str,
value))}")
#             else:
#                 entry_parts.append(f"{key}: {value}")

#         combined_text = "\n".join(entry_parts)
#         texts.append(combined_text)

#     return tokenizer(texts, truncation=True, padding="max_length",
max_length=256)

def tokenize_function(examples):
    prompts = []
    for i in range(len(examples["instruction"])):
        instruction = examples["instruction"][i]
        response = examples["response"][i]
        prompt_type = examples.get("prompt_type", ["analysis"] *
len(examples["instruction"]))[i]  # default to 'analysis'

        template = prompt_templates.get(prompt_type,
prompt_templates["analysis"])
        full_prompt = template.format(query=instruction) + "\n\
nAnswer: " + response
        prompts.append(full_prompt)

    return tokenizer(prompts, truncation=True, padding="max_length",
max_length=512)

def normalize_entry(entry):
    normalized = {}
    for key, value in entry.items():
        if isinstance(value, dict):
            for subkey, subval in value.items():
                normalized[f"{key}.{subkey}"] = str(subval) if subval
is not None else ""
        elif isinstance(value, list):
            normalized[key] = ", ".join(map(str, value))
        elif value is None:
```

```
            normalized[key] = ""
        else:
            normalized[key] = str(value)
    return normalized

# Normalize each entry
train_data_clean = [normalize_entry(entry) for entry in
split_data['train']]
test_data_clean = [normalize_entry(entry) for entry in
split_data['test']]

train_dataset_hf = Dataset.from_list(train_data_clean)
test_dataset_hf = Dataset.from_list(test_data_clean)
```

Prompt Engineering

```
# Define various prompting templates
prompt_templates = {
    "analysis": (
        "As a policy analyst, analyze the following policy issue:\n"
        "{query}\n\n"
        "Consider relevant socioeconomic factors, provide statistical
insights, "
        "and offer evidence-based recommendations."
    ),
    "comparative": (
        "As a policy analyst, compare these policy approaches:\n"
        "{query}\n\n"
        "Evaluate each using statistical data, consider implementation
challenges, "
        "and assess likely outcomes across different demographics."
    ),
    "forecast": (
        "As a policy analyst, forecast the outcomes of this policy
change:\n"
        "{query}\n\n"
        "Project short and long-term impacts, identify potential
unintended consequences, "
        "and quantify likely effects where possible."
    ),
}
```

Train & Test - Tokenization

```
tokenized_train = train_dataset_hf.map(tokenize_function,
batched=True)
tokenized_train.set_format(type="torch")
print("Tokenization complete with all features.")
```

{"model_id":"99ebce3462374091a0cf8cfe4bbab965","version_major":2,"version_minor":0}

```
Tokenization complete with all features.
```

```
tokenized_test = test_dataset_hf.map(tokenize_function, batched=True)
tokenized_test.set_format(type="torch")
print("Tokenization complete with all features.")
```

{"model_id":"f1a54043078c4033b52832f79b022e31","version_major":2,"version_minor":0}

```
Tokenization complete with all features.
```

Configer - PEFT, LoRA & QLoRA

```
lora_config = LoraConfig(
    r=8,
    lora_alpha=16,
    target_modules=['q_proj', 'k_proj', 'v_proj', 'o_proj'],
    lora_dropout=0.15,
    bias="none",
    task_type="CAUSAL_LM"
)

base_model.gradient_checkpointing_enable()
base_model = prepare_model_for_kbit_training(base_model)

peft_model = get_peft_model(base_model, lora_config)
peft_model.config.use_cache = False

print("After PEFT wrapping:")
print(trainable_parameters(peft_model))
```

```
After PEFT wrapping:
- Trainable model parameters: 4587520.
- All model parameters: 1808051200.
- Percentage of trainable model parameters: 0.25%
```

Train PH-Llama-3.1 Model & Evaluation

```
import torch
import os
data_collator = DataCollatorForLanguageModeling(tokenizer=tokenizer,
mlm=False)

os.environ["PYTORCH_CUDA_ALLOC_CONF"] = "expandable_segments:True"
```

```python
training_args = TrainingArguments(
    output_dir="./SocioLens-llama-3.2-3B",
    overwrite_output_dir=True,
    per_device_train_batch_size=4,
    per_device_eval_batch_size=4,
    gradient_accumulation_steps=4,  # Increased
    optim="adamw_8bit",
    num_train_epochs=8,  # Increased
    eval_strategy="steps",
    eval_steps=50,
    save_strategy="steps",
    save_steps=50,
    greater_is_better=False,
    logging_steps=1,
    weight_decay=0.01,  # Reduced
    warmup_steps=100,  # Increased
    logging_strategy="steps",
    learning_rate=4e-5,  # Slightly adjusted
    fp16=not torch.cuda.is_bf16_supported(),
    bf16=torch.cuda.is_bf16_supported(),
    lr_scheduler_type='cosine',
    seed=3407,
    group_by_length=True,
    max_grad_norm=1.0,
    gradient_checkpointing=True,
    report_to="wandb"
)

# training_args = TrainingArguments(
#     output_dir="./SocioLens-llama-3.2-3B",
#     overwrite_output_dir=True,
#     per_device_train_batch_size=4,                    # Increased
batch size
#     per_device_eval_batch_size=4,
#     gradient_accumulation_steps=4,                    # Effective
batch size = 4 * 4 = 16
#     optim="adamw_8bit",                               # Use 8-bit
AdamW
#     num_train_epochs=5,
#     eval_strategy="steps",
#     eval_steps=50,
#     save_strategy="steps",
#     save_steps=50,
#     greater_is_better=False,
#     logging_steps=1,
#     weight_decay=0.01,                                # Increased
weight decay
#     warmup_steps=50,                                  # Increased
```

```
warmup steps
#     logging_strategy="steps",
#     learning_rate=5e-5,                                  # Lower
learning rate
#     fp16=not torch.cuda.is_bf16_supported(),
#     bf16=torch.cuda.is_bf16_supported(),
#     lr_scheduler_type='cosine',                          # Use cosine
scheduler
#     seed=3407,
#     group_by_length=True,
#     max_grad_norm=1.0,                                   # Gradient
clipping
#     gradient_checkpointing=True,                         # Save memory
#     report_to="wandb"
# )


trainer = SFTTrainer(
    model=peft_model,
    args=training_args,
    peft_config=lora_config,
    train_dataset=tokenized_train,
    eval_dataset=tokenized_test,
    data_collator=data_collator,
)

torch.cuda.empty_cache() # Force Clear Cache Before Training

print("Starting training...")
trainer.train()
print(f"Training complete.")
```


```
{"model_id":"e6001ee831ca42c79bd2a9a6d300f397","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"42d997fec38747969e7a8de43863068b","version_major":2,"vers
ion_minor":0}
```


```
No label_names provided for model class `PeftModelForCausalLM`. Since
`PeftModel` hides base models input arguments, if label_names is not
given, label_names can't be set automatically within `Trainer`. Note
that empty label_names list will be used instead.
```

```
Starting training...
```

```
<IPython.core.display.HTML object>
```

```
eval_results = trainer.evaluate()
print("Evaluation Results:")
print(eval_results)
```

```
<IPython.core.display.HTML object>

Evaluation Results:
{'eval_loss': 0.8741981983184814, 'eval_runtime': 76.9247,
'eval_samples_per_second': 3.159, 'eval_steps_per_second': 0.793}

peft_model.config.save_pretrained("./SocioLens-llama-3.2-3B")

!ls -la ./SocioLens-llama-3.2-3B

total 32
drwxr-xr-x 7 student student 4096 Apr 26 02:19 .
drwxr-xr-x 9 student student 4096 Apr 26 02:18 ..
drwxr-xr-x 2 student student 4096 Apr 26 01:09 checkpoint-100
drwxr-xr-x 2 student student 4096 Apr 26 01:34 checkpoint-150
drwxr-xr-x 2 student student 4096 Apr 26 01:59 checkpoint-200
drwxr-xr-x 2 student student 4096 Apr 26 02:18 checkpoint-240
drwxr-xr-x 2 student student 4096 Apr 26 00:45 checkpoint-50
-rw-r--r-- 1 student student 1361 Apr 26 02:19 config.json

files = os.listdir("./SocioLens-llama-3.2-3B")
print("Files in the output directory:", files)

Files in the output directory: ['checkpoint-240', 'checkpoint-150',
'checkpoint-100', 'checkpoint-200', 'checkpoint-50', 'config.json']
```

Generate Text by Trained Model

```
import re
import random
from datetime import datetime

def generate_alpaca_text(
    prompt,
    max_length=512,
    temperature=0.0,
    top_p=0.95,
    system_message="You are SocioLens, an expert AI assistant
specializing in adult education policy, delivering concise, accurate,
and professional responses.",
    use_few_shot=True,
    use_cot=False,
    tokenizer=None,
    model=None,
    do_sample=False,
    user_id=None
):
    """
    Generates text using an Alpaca-style prompt format with varied,
```

```
professional conversational responses
    for common prompts and advanced prompt engineering for complex
tasks, using a single prompt input.

    :param prompt: The user input, containing the instruction or
question.
    :param max_length: The maximum length of the generated text.
    :param temperature: Sampling temperature for controlling
randomness.
    :param top_p: Nucleus sampling parameter for controlling
creativity.
    :param system_message: System message to define the model's role
or persona.
    :param use_few_shot: Whether to include few-shot examples in the
prompt.
    :param use_cot: Whether to encourage chain-of-thought reasoning.
    :param tokenizer: The tokenizer for the model.
    :param model: The fine-tuned model for text generation.
    :param do_sample: Whether to use sampling or greedy decoding.
    :param user_id: Optional identifier for the user to ensure varied
responses across users.
    :return: A string containing the generated response.
    """
    # Validate inputs
    if not prompt:
        raise ValueError("Prompt cannot be empty.")
    if not tokenizer or not model:
        raise ValueError("Tokenizer and model must be provided.")

    # Set random seed for varied responses
    seed = hash(user_id) if user_id else
int(datetime.now().timestamp())
    random.seed(seed)

    # Response templates for conversational prompts
    conversational_templates = {
        r"^(hi|hello|hey|greetings)(\s.*)?$": {
            "greetings": ["Greetings", "Hello", "Good day"],
            "status": [
                "I'm performing optimally and ready to assist",
                "I'm fully operational and here to help",
                "I'm at peak performance and eager to support you"
            ],
            "offer": [
                "How may I help you today?",
                "What can I assist you with today?",
                "How can I support your needs today?"
            ],
            "combine": lambda g, s, o: f"{g}! {s}. {o}"
```

```
        },
        r"^how\s+are\s+you(\s*doing)?\?$": {
            "greetings": [""],
            "status": [
                "I'm functioning at peak performance and ready to
assist",
                "I'm operating smoothly and here to help",
                "I'm in optimal condition and eager to support"
            ],
            "offer": [
                "How about you—how may I support your needs today?",
                "What can I assist you with today?",
                "How may I help you today?"
            ],
            "combine": lambda g, s, o: f"{s}. {o}"
        },
        r"^who\s+are\s+you\?$": {
            "intro": [
                "I am SocioLens, an AI assistant specializing in adult
education policy",
                "I am SocioLens, an expert AI designed for adult
education policy",
                "I am SocioLens, your AI assistant for adult education
policy"
            ],
            "creators": ["developed by Yash, Shrestha, and Parin"],
            "offer": [
                "How can I assist you today?",
                "What can I help you with today?",
                "How may I support you today?"
            ],
            "combine": lambda i, c, o: f"{i}, {c}. {o}"
        },
        r"^tell\s+me\s+about\s+(you|yourself)(\?)?$": {
            "intro": [
                "I am SocioLens, a large language model",
                "I am SocioLens, an advanced AI",
                "I am SocioLens, a sophisticated language model"
            ],
            "creators": ["created by Yash, Shrestha, and Parin"],
            "purpose": [
                "I'm designed to provide accurate and insightful
answers, particularly in adult education policy",
                "My purpose is to deliver precise and professional
responses, especially on adult education policy",
                "I'm built to offer reliable and detailed insights,
focusing on adult education policy"
            ],
            "offer": [
```

```
                "What would you like to explore?",
                "What topic would you like to discuss?",
                "What can I help you learn about today?"
            ],
            "combine": lambda i, c, p, o: f"{i} {c}. {p}. {o}"
        }
    }

    # Check for conversational prompts
    prompt_lower = prompt.lower().strip()
    for pattern, template in conversational_templates.items():
        if re.match(pattern, prompt_lower):
            components = {
                key: random.choice(values)
                for key, values in template.items()
                if key != "combine"
            }
            response = template["combine"](*components.values())
            return response

    # Handle temperature and do_sample compatibility
    if temperature == 0.0:
        do_sample = False
    elif do_sample and temperature <= 0.0:
        temperature = 0.7

    # Updated few-shot examples for single prompt input
    few_shot_examples = [
        {
            "prompt": "Summarize the key findings of the latest adult
education policy research.",
            "response": (
                "Recent adult education policy research highlights
increased literacy rates and vocational skills development, "
                "particularly in underserved regions, leading to
improved employability and economic outcomes."
            )
        },
        {
            "prompt": "What are the socio-economic factors that affect
public health?",
            "response": (
                "While socio-economic factors like income, education,
and employment significantly impact public health, "
                "my expertise lies in adult education policy. Would
you like me to provide insights on how adult education "
                "can address these factors, or focus on a related
policy topic?"
            )
```

```
        }
    ] if use_few_shot else []

    # Construct few-shot examples section
    few_shot_prompt = ""
    if few_shot_examples:
        few_shot_prompt = "\n\n### Examples:\n"
        for example in few_shot_examples:
            few_shot_prompt += (
                f"#### Example Prompt:\n{example['prompt']}\n\n"
                f"#### Example Response:\n{example['response']}\n\n"
            )

    # Refined CoT prompt for single input
    cot_prompt = (
        "\nPlease reason step by step to ensure a clear and accurate
response. "
        "Focus on the prompt and provide a professional answer,
prioritizing adult education policy if relevant. "
        "If the prompt is outside this domain, acknowledge it and
offer to assist within my expertise."
    ) if use_cot else ""

    # Construct Alpaca-style prompt
    alpaca_prompt = (
        f"### System:\n{system_message}\n\n"
        "Below is a prompt that describes a task or question. "
        "Write a response that appropriately completes the request,
ensuring relevance to adult education policy when applicable.\n"
        f"{few_shot_prompt}"
        "### Prompt:\n"
        f"{prompt}{cot_prompt}\n\n"
        "### Response:\n"
    )

    # Check prompt length
    tokenized_prompt = tokenizer(alpaca_prompt, return_tensors="pt",
truncation=False)
    if tokenized_prompt.input_ids.size(1) >
tokenizer.model_max_length:
        raise ValueError("Prompt exceeds model's maximum context
length.")

    # Tokenize prompt
    inputs = tokenizer(alpaca_prompt, return_tensors="pt",
padding=True, truncation=True)
    inputs = {key: value.to(model.device) for key, value in
inputs.items()}

    # Generate output
```

```
    outputs = model.generate(
        input_ids=inputs["input_ids"],
        attention_mask=inputs["attention_mask"],
        max_length=max_length,
        do_sample=do_sample,
        temperature=temperature if do_sample else None,
        top_p=top_p if do_sample else None,
        pad_token_id=tokenizer.eos_token_id
    )

    # Decode generated text
    generated_text = tokenizer.decode(outputs[0],
skip_special_tokens=True)

    # Extract response part
    response_start = generated_text.find("### Response:") + len("###
Response:\n")
    if response_start != -1:
        generated_text = generated_text[response_start:].strip()

    return generated_text
```

Hi - Converstion with our LLM

```
response1 = generate_alpaca_text(
    prompt="hi",
    tokenizer=tokenizer,
    model=peft_model
)
print(response1)
```

```
Hello! I'm at peak performance and eager to support you. How may I
help you today?
```

Who are you? - Converstion with our LLM

```
response2 = generate_alpaca_text(
    prompt="who are you?",
    tokenizer=tokenizer,
    model=peft_model
)
print(response2)
```

```
I am SocioLens, an AI assistant specializing in adult education
policy, developed by Yash, Shrestha, and Parin. How can I assist you
today?
```

Tell me about you? - Converstion with our LLM

```
response3 = generate_alpaca_text(
    prompt="Tell me about you?",
    tokenizer=tokenizer,
    model=peft_model
)
print(response3)
```

```
I am SocioLens, a large language model created by Yash, Shrestha, and
Parin. I'm built to offer reliable and detailed insights, focusing on
adult education policy. What would you like to explore?
```

```
prompt = """U.S. Healthcare vs. Other High-Income Countries abstract
This report compares the quality of healthcare in the United States to
other high-income countries,
focusing on key metrics such as life expectancy, all-cause mortality,
maternal mortality, and premature death.
It discusses how high healthcare spending in the U.S. does not
translate into better outcomes."""


response4 = generate_alpaca_text(
    prompt=prompt,
    max_length=512,
    temperature=0.0,
    top_p=0.9,
    use_few_shot=True,
    use_cot=True,
    tokenizer=tokenizer,
    model=peft_model
)
print(response4)  # Output: (Model-generated summary, e.g., Recent
adult education policy research highlights significant improvements in
literacy and numerical skills...)
```

```
While the United States spends significantly more on healthcare than
other high-income countries, our outcomes lag behind, with higher all-
cause mortality, maternal mortality, and premature death. This report
highlights the need for more effective policies to address
socioeconomic disparities and improve healthcare quality.
```

```
prompt = "What is the policy impact of adult education on food
security?"

response5 = generate_alpaca_text(
    prompt=prompt,
    max_length=512,
    temperature=0.0,
    top_p=0.9,
    use_few_shot=True,
```

```
    use_cot=True,
    tokenizer=tokenizer,
    model=peft_model
)
print(response5)

#While adult education can improve literacy and vocational skills, its
direct impact on food security is limited.
#However, it indirectly enhances employment and economic stability,
reducing food insecurity through increased
#income and stability. Considerable research suggests that adult
education programs, when integrated with food
#security initiatives, can provide additional support, but the
specific policy implications are nuanced and
#require further study.
```

```
While adult education can improve literacy and vocational skills, its
direct impact on food security is limited. However, it indirectly
enhances employment and economic stability, reducing food insecurity
through increased income and stability. Considerable research suggests
that adult education programs, when integrated with food security
initiatives, can provide additional support, but the specific policy
implications are nuanced and require further study.
```

```
prompt = "Recent policy interventions in adult education aim to
improve literacy and numerical skills. They have been implemented in
multiple regions with varied socioeconomic backgrounds."

response5 = generate_alpaca_text(
    prompt=prompt,
    max_length=512,
    temperature=0.0,
    top_p=0.95,
    use_few_shot=True,
    use_cot=True,
    tokenizer=tokenizer,
    model=peft_model
)
print(response5)
```

```
While socioeconomic factors like income, education, and employment
influence public health, adult education policy focuses on literacy
and numeracy, directly addressing these factors. In regions with
higher poverty, literacy interventions can reduce health disparities
by 10-15 percent. In wealthier areas, vocational training programs,
like those in the Northeast, enhance employability and reduce health
risks. However, in rural areas with limited infrastructure,
transportation, and access to healthcare, literacy and numeracy
programs can mitigate 20-25 percent of health disparities. These
```

```
regional variations highlight the importance of tailored policy
interventions.

# Save your fine-tuned model to a local directory
model_save_path = "./SocioLens-llama-3.2-3B"
trainer.save_model(model_save_path)
tokenizer.save_pretrained(model_save_path)

torch.save(peft_model.state_dict(), "./model/SocioLens-llama-3.2-
3B.pth")

from huggingface_hub import HfApi, HfFolder, Repository

from huggingface_hub import login
login(token="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")

trainer.push_to_hub("iyashnayi/SocioLens-llama-3.2-3B")
```