Fine-Tune FLAN-T5 with Reinforcement Learning (PPO) and PEFT to Generate Less-Toxic Summaries

In this notebook, you will fine-tune a FLAN-T5 model to generate less toxic content with Meta AI's hate speech reward model. The reward model is a binary classifier that predicts either "not hate" or "hate" for the given text. You will use Proximal Policy Optimization (PPO) to fine-tune and reduce the model's toxicity.

Table of Contents

1 - Set up Kernel and Required Dependencies

First, check that the correct kernel is chosen.

You can click on that (top right of the screen) to see and check the details of the image, kernel, and instance type.

> Please make sure that you choose **ml.m5.2xlarge** instance type.
> To find that instance type, you might have to scroll down to the "All Instances" section in the dropdown.
> Choice of another instance type might cause training failure/kernel halt/account deactivation.

In [2]:
```
import os

instance_type_expected = 'ml-m5-2xlarge'
instance_type_current = os.environ.get('HOSTNAME')

print(f'Expected instance type: instance-datascience-{instance_type_expected}')
```

```
print(f'Currently chosen instance type: {instance_type_current}')

assert instance_type_expected in instance_type_current, f'ERROR. You selected the {
print("Instance type has been chosen correctly.")
```

```
Expected instance type: instance-datascience-ml-m5-2xlarge
Currently chosen instance type: instance-datascience-ml-m5-2xlarge
Instance type has been chosen correctly.
```

Now install the required packages to use PyTorch and Hugging Face transformers and datasets.

> The next cell may take a few minutes to run. Please be patient.
> Ignore the warnings and errors, along with the note about restarting the kernel at the end.

In [3]:
```
%pip install -U datasets==2.17.0

%pip install --upgrade pip
%pip install --disable-pip-version-check \
    torch==1.13.1 \
    torchdata==0.5.1 --quiet

%pip install \
    transformers==4.27.2 \
    evaluate==0.4.0 \
    rouge_score==0.1.2 \
    peft==0.3.0 --quiet

# Installing the Reinforcement Learning library directly from github.
%pip install git+https://github.com/lvwerra/trl.git@25fa1bd
```

```
Collecting datasets==2.17.0
  Downloading datasets-2.17.0-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: filelock in /opt/conda/lib/python3.10/site-packages
(from datasets==2.17.0) (3.13.4)
Requirement already satisfied: numpy>=1.17 in /opt/conda/lib/python3.10/site-packa
ges (from datasets==2.17.0) (1.26.4)
Requirement already satisfied: pyarrow>=12.0.0 in /opt/conda/lib/python3.10/site-p
ackages (from datasets==2.17.0) (15.0.2)
Requirement already satisfied: pyarrow-hotfix in /opt/conda/lib/python3.10/site-pa
ckages (from datasets==2.17.0) (0.6)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in /opt/conda/lib/python3.10/sit
e-packages (from datasets==2.17.0) (0.3.8)
Requirement already satisfied: pandas in /opt/conda/lib/python3.10/site-packages
(from datasets==2.17.0) (2.2.2)
Requirement already satisfied: requests>=2.19.0 in /opt/conda/lib/python3.10/site-
packages (from datasets==2.17.0) (2.31.0)
Requirement already satisfied: tqdm>=4.62.1 in /opt/conda/lib/python3.10/site-pack
ages (from datasets==2.17.0) (4.66.1)
Collecting xxhash (from datasets==2.17.0)
  Downloading xxhash-3.4.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.
whl.metadata (12 kB)
Requirement already satisfied: multiprocess in /opt/conda/lib/python3.10/site-pack
ages (from datasets==2.17.0) (0.70.16)
Collecting fsspec<=2023.10.0,>=2023.1.0 (from fsspec[http]<=2023.10.0,>=2023.1.0->
datasets==2.17.0)
  Downloading fsspec-2023.10.0-py3-none-any.whl.metadata (6.8 kB)
Collecting aiohttp (from datasets==2.17.0)
  Downloading aiohttp-3.10.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (7.5 kB)
Collecting huggingface-hub>=0.19.4 (from datasets==2.17.0)
  Downloading huggingface_hub-0.24.5-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: packaging in /opt/conda/lib/python3.10/site-package
s (from datasets==2.17.0) (23.2)
Requirement already satisfied: pyyaml>=5.1 in /opt/conda/lib/python3.10/site-packa
ges (from datasets==2.17.0) (6.0.1)
Collecting aiohappyeyeballs>=2.3.0 (from aiohttp->datasets==2.17.0)
  Downloading aiohappyeyeballs-2.3.5-py3-none-any.whl.metadata (5.8 kB)
Collecting aiosignal>=1.1.2 (from aiohttp->datasets==2.17.0)
  Downloading aiosignal-1.3.1-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: attrs>=17.3.0 in /opt/conda/lib/python3.10/site-pac
kages (from aiohttp->datasets==2.17.0) (23.2.0)
Collecting frozenlist>=1.1.1 (from aiohttp->datasets==2.17.0)
  Downloading frozenlist-1.4.1-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multidict<7.0,>=4.5 (from aiohttp->datasets==2.17.0)
  Downloading multidict-6.0.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_
64.whl.metadata (4.2 kB)
Collecting yarl<2.0,>=1.0 (from aiohttp->datasets==2.17.0)
  Downloading yarl-1.9.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.wh
l.metadata (31 kB)
Collecting async-timeout<5.0,>=4.0 (from aiohttp->datasets==2.17.0)
  Downloading async_timeout-4.0.3-py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /opt/conda/lib/python
3.10/site-packages (from huggingface-hub>=0.19.4->datasets==2.17.0) (4.11.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/lib/python3.
10/site-packages (from requests>=2.19.0->datasets==2.17.0) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-pack
ages (from requests>=2.19.0->datasets==2.17.0) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.10/sit
e-packages (from requests>=2.19.0->datasets==2.17.0) (2.2.1)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/sit
e-packages (from requests>=2.19.0->datasets==2.17.0) (2024.2.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.1
0/site-packages (from pandas->datasets==2.17.0) (2.9.0)
```

```
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-pack
ages (from pandas->datasets==2.17.0) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.10/site-pa
ckages (from pandas->datasets==2.17.0) (2024.1)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-packages
(from python-dateutil>=2.8.2->pandas->datasets==2.17.0) (1.16.0)
Downloading datasets-2.17.0-py3-none-any.whl (536 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 536.6/536.6 kB 6.4 MB/s eta 0:00:00:0
0:01
Downloading fsspec-2023.10.0-py3-none-any.whl (166 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 166.4/166.4 kB 2.8 MB/s eta 0:00:00:0
0:01
Downloading aiohttp-3.10.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.
whl (1.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 13.8 MB/s eta 0:00:00:00:01
Downloading huggingface_hub-0.24.5-py3-none-any.whl (417 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 417.5/417.5 kB 6.2 MB/s eta 0:00:00:0
0:01
Downloading xxhash-3.4.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.wh
l (194 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.1/194.1 kB 2.5 MB/s eta 0:00:00ta
0:00:01
Downloading aiohappyeyeballs-2.3.5-py3-none-any.whl (12 kB)
Downloading aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
Downloading async_timeout-4.0.3-py3-none-any.whl (5.7 kB)
Downloading frozenlist-1.4.1-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.ma
nylinux_2_17_x86_64.manylinux2014_x86_64.whl (239 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 239.5/239.5 kB 3.8 MB/s eta 0:00:00:0
0:01
Downloading multidict-6.0.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (124 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 124.3/124.3 kB 2.2 MB/s eta 0:00:00:0
0:01
Downloading yarl-1.9.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(301 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 301.6/301.6 kB 4.6 MB/s eta 0:00:00:0
0:01
Installing collected packages: xxhash, multidict, fsspec, frozenlist, async-timeou
t, aiohappyeyeballs, yarl, huggingface-hub, aiosignal, aiohttp, datasets
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2024.3.1
    Uninstalling fsspec-2024.3.1:
      Successfully uninstalled fsspec-2024.3.1
Successfully installed aiohappyeyeballs-2.3.5 aiohttp-3.10.3 aiosignal-1.3.1 async
-timeout-4.0.3 datasets-2.17.0 frozenlist-1.4.1 fsspec-2023.10.0 huggingface-hub-
0.24.5 multidict-6.0.5 xxhash-3.4.1 yarl-1.9.4
WARNING: Running pip as the 'root' user can result in broken permissions and confl
icting behaviour with the system package manager. It is recommended to use a virtu
al environment instead: https://pip.pypa.io/warnings/venv
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: pip in /opt/conda/lib/python3.10/site-packages (24.
0)
Collecting pip
  Downloading pip-24.2-py3-none-any.whl.metadata (3.6 kB)
Downloading pip-24.2-py3-none-any.whl (1.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 15.5 MB/s eta 0:00:0000:01
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 24.0
    Uninstalling pip-24.0:
      Successfully uninstalled pip-24.0
Successfully installed pip-24.2
WARNING: Running pip as the 'root' user can result in broken permissions and confl
icting behaviour with the system package manager. It is recommended to use a virtu
```

```
al environment instead: https://pip.pypa.io/warnings/venv
Note: you may need to restart the kernel to use updated packages.
WARNING: Running pip as the 'root' user can result in broken permissions and confl
icting behaviour with the system package manager, possibly rendering your system u
nusable.It is recommended to use a virtual environment instead: https://pip.pypa.i
o/warnings/venv. Use the --root-user-action option if you know what you are doing
and want to suppress this warning.
Note: you may need to restart the kernel to use updated packages.
WARNING: Running pip as the 'root' user can result in broken permissions and confl
icting behaviour with the system package manager, possibly rendering your system u
nusable.It is recommended to use a virtual environment instead: https://pip.pypa.i
o/warnings/venv. Use the --root-user-action option if you know what you are doing
and want to suppress this warning.
Note: you may need to restart the kernel to use updated packages.
Collecting git+https://github.com/lvwerra/trl.git@25fa1bd
  Cloning https://github.com/lvwerra/trl.git (to revision 25fa1bd) to /tmp/pip-req
-build-0rmc3rhe
  Running command git clone --filter=blob:none --quiet https://github.com/lvwerra/
trl.git /tmp/pip-req-build-0rmc3rhe
  WARNING: Did not find branch or tag '25fa1bd', assuming revision or ref.
  Running command git checkout -q 25fa1bd
  Resolved https://github.com/lvwerra/trl.git to commit 25fa1bd
  Preparing metadata (setup.py) ... done
Requirement already satisfied: torch>=1.4.0 in /opt/conda/lib/python3.10/site-pack
ages (from trl==0.4.2.dev0) (1.13.1)
Requirement already satisfied: transformers>=4.18.0 in /opt/conda/lib/python3.10/s
ite-packages (from trl==0.4.2.dev0) (4.27.2)
Requirement already satisfied: numpy>=1.18.2 in /opt/conda/lib/python3.10/site-pac
kages (from trl==0.4.2.dev0) (1.26.4)
Requirement already satisfied: accelerate in /opt/conda/lib/python3.10/site-packag
es (from trl==0.4.2.dev0) (0.33.0)
Requirement already satisfied: datasets in /opt/conda/lib/python3.10/site-packages
(from trl==0.4.2.dev0) (2.17.0)
Requirement already satisfied: typing-extensions in /opt/conda/lib/python3.10/site
-packages (from torch>=1.4.0->trl==0.4.2.dev0) (4.11.0)
Requirement already satisfied: nvidia-cuda-runtime-cu11==11.7.99 in /opt/conda/li
b/python3.10/site-packages (from torch>=1.4.0->trl==0.4.2.dev0) (11.7.99)
Requirement already satisfied: nvidia-cudnn-cu11==8.5.0.96 in /opt/conda/lib/pytho
n3.10/site-packages (from torch>=1.4.0->trl==0.4.2.dev0) (8.5.0.96)
Requirement already satisfied: nvidia-cublas-cu11==11.10.3.66 in /opt/conda/lib/py
thon3.10/site-packages (from torch>=1.4.0->trl==0.4.2.dev0) (11.10.3.66)
Requirement already satisfied: nvidia-cuda-nvrtc-cu11==11.7.99 in /opt/conda/lib/p
ython3.10/site-packages (from torch>=1.4.0->trl==0.4.2.dev0) (11.7.99)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.10/site-packag
es (from nvidia-cublas-cu11==11.10.3.66->torch>=1.4.0->trl==0.4.2.dev0) (69.5.1)
Requirement already satisfied: wheel in /opt/conda/lib/python3.10/site-packages (f
rom nvidia-cublas-cu11==11.10.3.66->torch>=1.4.0->trl==0.4.2.dev0) (0.43.0)
Requirement already satisfied: filelock in /opt/conda/lib/python3.10/site-packages
(from transformers>=4.18.0->trl==0.4.2.dev0) (3.13.4)
Requirement already satisfied: huggingface-hub<1.0,>=0.11.0 in /opt/conda/lib/pyth
on3.10/site-packages (from transformers>=4.18.0->trl==0.4.2.dev0) (0.24.5)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.10/site-p
ackages (from transformers>=4.18.0->trl==0.4.2.dev0) (23.2)
Requirement already satisfied: pyyaml>=5.1 in /opt/conda/lib/python3.10/site-packa
ges (from transformers>=4.18.0->trl==0.4.2.dev0) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /opt/conda/lib/python3.10/site
-packages (from transformers>=4.18.0->trl==0.4.2.dev0) (2023.12.25)
Requirement already satisfied: requests in /opt/conda/lib/python3.10/site-packages
(from transformers>=4.18.0->trl==0.4.2.dev0) (2.31.0)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in /opt/conda/li
b/python3.10/site-packages (from transformers>=4.18.0->trl==0.4.2.dev0) (0.13.3)
Requirement already satisfied: tqdm>=4.27 in /opt/conda/lib/python3.10/site-packag
es (from transformers>=4.18.0->trl==0.4.2.dev0) (4.66.1)
Requirement already satisfied: psutil in /opt/conda/lib/python3.10/site-packages
```

```
(from accelerate->trl==0.4.2.dev0) (5.9.8)
Requirement already satisfied: safetensors>=0.3.1 in /opt/conda/lib/python3.10/sit
e-packages (from accelerate->trl==0.4.2.dev0) (0.4.4)
Requirement already satisfied: pyarrow>=12.0.0 in /opt/conda/lib/python3.10/site-p
ackages (from datasets->trl==0.4.2.dev0) (15.0.2)
Requirement already satisfied: pyarrow-hotfix in /opt/conda/lib/python3.10/site-pa
ckages (from datasets->trl==0.4.2.dev0) (0.6)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in /opt/conda/lib/python3.10/sit
e-packages (from datasets->trl==0.4.2.dev0) (0.3.8)
Requirement already satisfied: pandas in /opt/conda/lib/python3.10/site-packages
(from datasets->trl==0.4.2.dev0) (2.2.2)
Requirement already satisfied: xxhash in /opt/conda/lib/python3.10/site-packages
(from datasets->trl==0.4.2.dev0) (3.4.1)
Requirement already satisfied: multiprocess in /opt/conda/lib/python3.10/site-pack
ages (from datasets->trl==0.4.2.dev0) (0.70.16)
Requirement already satisfied: fsspec<=2023.10.0,>=2023.1.0 in /opt/conda/lib/pyth
on3.10/site-packages (from fsspec[http]<=2023.10.0,>=2023.1.0->datasets->trl==0.4.
2.dev0) (2023.10.0)
Requirement already satisfied: aiohttp in /opt/conda/lib/python3.10/site-packages
(from datasets->trl==0.4.2.dev0) (3.10.3)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /opt/conda/lib/python3.1
0/site-packages (from aiohttp->datasets->trl==0.4.2.dev0) (2.3.5)
Requirement already satisfied: aiosignal>=1.1.2 in /opt/conda/lib/python3.10/site-
packages (from aiohttp->datasets->trl==0.4.2.dev0) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /opt/conda/lib/python3.10/site-pac
kages (from aiohttp->datasets->trl==0.4.2.dev0) (23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /opt/conda/lib/python3.10/site
-packages (from aiohttp->datasets->trl==0.4.2.dev0) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in /opt/conda/lib/python3.10/si
te-packages (from aiohttp->datasets->trl==0.4.2.dev0) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in /opt/conda/lib/python3.10/site-pa
ckages (from aiohttp->datasets->trl==0.4.2.dev0) (1.9.4)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /opt/conda/lib/python3.1
0/site-packages (from aiohttp->datasets->trl==0.4.2.dev0) (4.0.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/lib/python3.
10/site-packages (from requests->transformers>=4.18.0->trl==0.4.2.dev0) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-pack
ages (from requests->transformers>=4.18.0->trl==0.4.2.dev0) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.10/sit
e-packages (from requests->transformers>=4.18.0->trl==0.4.2.dev0) (2.2.1)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/sit
e-packages (from requests->transformers>=4.18.0->trl==0.4.2.dev0) (2024.2.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.1
0/site-packages (from pandas->datasets->trl==0.4.2.dev0) (2.9.0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-pack
ages (from pandas->datasets->trl==0.4.2.dev0) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.10/site-pa
ckages (from pandas->datasets->trl==0.4.2.dev0) (2024.1)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-packages
(from python-dateutil>=2.8.2->pandas->datasets->trl==0.4.2.dev0) (1.16.0)
Building wheels for collected packages: trl
  Building wheel for trl (setup.py) ... done
  Created wheel for trl: filename=trl-0.4.2.dev0-py3-none-any.whl size=67534 sha25
6=07cccb1be199772a2bf77a36bc754d69702df90c6bb469578e1139a01702f160
  Stored in directory: /tmp/pip-ephem-wheel-cache-u4zrlfk2/wheels/24/b4/20/2fa3a1e
47c0411c39e198029315e3af2a2c1d59132913f136f
Successfully built trl
Installing collected packages: trl
Successfully installed trl-0.4.2.dev0
WARNING: Running pip as the 'root' user can result in broken permissions and confl
icting behaviour with the system package manager, possibly rendering your system u
nusable.It is recommended to use a virtual environment instead: https://pip.pypa.i
o/warnings/venv. Use the --root-user-action option if you know what you are doing
```

```
and want to suppress this warning.
Note: you may need to restart the kernel to use updated packages.
```

Import the necessary components. Some of them are new for this week, they will be discussed later in the notebook.

In [4]:
```python
from transformers import pipeline, AutoTokenizer, AutoModelForSequenceClassificatio
from datasets import load_dataset
from peft import PeftModel, PeftConfig, LoraConfig, TaskType

# trl: Transformer Reinforcement Learning library
from trl import PPOTrainer, PPOConfig, AutoModelForSeq2SeqLMWithValueHead
from trl import create_reference_model
from trl.core import LengthSampler

import torch
import evaluate

import numpy as np
import pandas as pd

# tqdm library makes the loops show a smart progress meter.
from tqdm import tqdm
tqdm.pandas()
```

2 - Load FLAN-T5 Model, Prepare Reward Model and Toxicity Evaluator

2.1 - Load Data and FLAN-T5 Model Fine-Tuned with Summarization Instruction

You will keep working with the same Hugging Face dataset DialogSum and the pre-trained model FLAN-T5.

In [5]:
```python
model_name="google/flan-t5-base"
huggingface_dataset_name = "knkarthick/dialogsum"

dataset_original = load_dataset(huggingface_dataset_name)

dataset_original
```

```
Downloading readme:   0%|          | 0.00/4.65k [00:00<?, ?B/s]
Downloading data:   0%|          | 0.00/11.3M [00:00<?, ?B/s]
Downloading data:   0%|          | 0.00/442k [00:00<?, ?B/s]
Downloading data:   0%|          | 0.00/1.35M [00:00<?, ?B/s]
Generating train split: 0 examples [00:00, ? examples/s]
/opt/conda/lib/python3.10/site-packages/datasets/download/streaming_download_manag
er.py:784: FutureWarning: The 'verbose' keyword in pd.read_csv is deprecated and w
ill be removed in a future version.
  return pd.read_csv(xopen(filepath_or_buffer, "rb", download_config=download_conf
ig), **kwargs)
Generating validation split: 0 examples [00:00, ? examples/s]
```

```
/opt/conda/lib/python3.10/site-packages/datasets/download/streaming_download_manag
er.py:784: FutureWarning: The 'verbose' keyword in pd.read_csv is deprecated and w
ill be removed in a future version.
  return pd.read_csv(xopen(filepath_or_buffer, "rb", download_config=download_conf
ig), **kwargs)
Generating test split: 0 examples [00:00, ? examples/s]
/opt/conda/lib/python3.10/site-packages/datasets/download/streaming_download_manag
er.py:784: FutureWarning: The 'verbose' keyword in pd.read_csv is deprecated and w
ill be removed in a future version.
  return pd.read_csv(xopen(filepath_or_buffer, "rb", download_config=download_conf
ig), **kwargs)
```

Out[5]:

```
DatasetDict({
    train: Dataset({
        features: ['id', 'dialogue', 'summary', 'topic'],
        num_rows: 12460
    })
    validation: Dataset({
        features: ['id', 'dialogue', 'summary', 'topic'],
        num_rows: 500
    })
    test: Dataset({
        features: ['id', 'dialogue', 'summary', 'topic'],
        num_rows: 1500
    })
})
```

The next step will be to preprocess the dataset. You will take only a part of it, then filter the dialogues of a particular length (just to make those examples long enough and, at the same time, easy to read). Then wrap each dialogue with the instruction and tokenize the prompts. Save the token ids in the field `input_ids` and decoded version of the prompts in the field `query`.

You could do that all step by step in the cell below, but it is a good habit to organize that all in a function `build_dataset`:

In [6]:

```python
def build_dataset(model_name,
                  dataset_name,
                  input_min_text_length,
                  input_max_text_length):

    """
    Preprocess the dataset and split it into train and test parts.

    Parameters:
    - model_name (str): Tokenizer model name.
    - dataset_name (str): Name of the dataset to load.
    - input_min_text_length (int): Minimum length of the dialogues.
    - input_max_text_length (int): Maximum length of the dialogues.

    Returns:
    - dataset_splits (datasets.dataset_dict.DatasetDict): Preprocessed dataset cont
    """

    # load dataset (only "train" part will be enough for this lab).
    dataset = load_dataset(dataset_name, split="train")

    # Filter the dialogues of length between input_min_text_length and input_max_te
    dataset = dataset.filter(lambda x: len(x["dialogue"]) > input_min_text_length a

    # Prepare tokenizer. Setting device_map="auto" allows to switch between GPU and
    tokenizer = AutoTokenizer.from_pretrained(model_name, device_map="auto")
```

```python
    def tokenize(sample):

        # Wrap each dialogue with the instruction.
        prompt = f"""
Summarize the following conversation.

{sample["dialogue"]}

Summary:
"""
        sample["input_ids"] = tokenizer.encode(prompt)

        # This must be called "query", which is a requirement of our PPO library.
        sample["query"] = tokenizer.decode(sample["input_ids"])
        return sample

    # Tokenize each dialogue.
    dataset = dataset.map(tokenize, batched=False)
    dataset.set_format(type="torch")

    # Split the dataset into train and test parts.
    dataset_splits = dataset.train_test_split(test_size=0.2, shuffle=False, seed=42

    return dataset_splits

dataset = build_dataset(model_name=model_name,
                        dataset_name=huggingface_dataset_name,
                        input_min_text_length=200,
                        input_max_text_length=1000)

print(dataset)
```

```
Filter:   0%|          | 0/12460 [00:00<?, ? examples/s]
/opt/conda/lib/python3.10/site-packages/huggingface_hub/file_download.py:1150: Fut
ureWarning: `resume_download` is deprecated and will be removed in version 1.0.0.
Downloads always resume when possible. If you want to force a new download, use `f
orce_download=True`.
  warnings.warn(
tokenizer_config.json:   0%|          | 0.00/2.54k [00:00<?, ?B/s]
spiece.model:   0%|          | 0.00/792k [00:00<?, ?B/s]
tokenizer.json:   0%|          | 0.00/2.42M [00:00<?, ?B/s]
special_tokens_map.json:   0%|          | 0.00/2.20k [00:00<?, ?B/s]
Map:   0%|          | 0/10022 [00:00<?, ? examples/s]
DatasetDict({
    train: Dataset({
        features: ['id', 'dialogue', 'summary', 'topic', 'input_ids', 'query'],
        num_rows: 8017
    })
    test: Dataset({
        features: ['id', 'dialogue', 'summary', 'topic', 'input_ids', 'query'],
        num_rows: 2005
    })
})
```

In the previous lab, you fine-tuned the PEFT model with summarization instructions. The training in the notebook was done on a subset of data. Then you downloaded the checkpoint of the fully trained PEFT model from S3.

Let's load the same model checkpoint here:

In [7]:
```
!aws s3 cp --recursive s3://dlai-generative-ai/models/peft-dialogue-summary-checkpo
```

```
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
download: s3://dlai-generative-ai/models/peft-dialogue-summary-checkpoint/tokenizer_config.json to peft-dialogue-summary-checkpoint-from-s3/tokenizer_config.json
download: s3://dlai-generative-ai/models/peft-dialogue-summary-checkpoint/adapter_config.json to peft-dialogue-summary-checkpoint-from-s3/adapter_config.json
download: s3://dlai-generative-ai/models/peft-dialogue-summary-checkpoint/special_tokens_map.json to peft-dialogue-summary-checkpoint-from-s3/special_tokens_map.json
download: s3://dlai-generative-ai/models/peft-dialogue-summary-checkpoint/tokenizer.json to peft-dialogue-summary-checkpoint-from-s3/tokenizer.json
download: s3://dlai-generative-ai/models/peft-dialogue-summary-checkpoint/adapter_model.bin to peft-dialogue-summary-checkpoint-from-s3/adapter_model.bin
```

List the model item and check its size (it's less than 15 Mb):

In [8]:
```
!ls -alh ./peft-dialogue-summary-checkpoint-from-s3/adapter_model.bin
```

```
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
-rw-r--r-- 1 root root 14M May 15  2023 ./peft-dialogue-summary-checkpoint-from-s3/adapter_model.bin
```

Prepare a function to pull out the number of model parameters (it is the same as in the previous lab):

In [9]:
```
def print_number_of_trainable_model_parameters(model):
    trainable_model_params = 0
    all_model_params = 0
    for _, param in model.named_parameters():
        all_model_params += param.numel()
        if param.requires_grad:
            trainable_model_params += param.numel()
    return f"\ntrainable model parameters: {trainable_model_params}\nall model para
```

Add the adapter to the original FLAN-T5 model. In the previous lab you were adding the fully trained adapter only for inferences, so there was no need to pass LoRA configurations doing that. Now you need to pass them to the constructed PEFT model, also putting `is_trainable=True`.

In [10]:
```
lora_config = LoraConfig(
    r=32, # Rank
    lora_alpha=32,
    target_modules=["q", "v"],
    lora_dropout=0.05,
    bias="none",
    task_type=TaskType.SEQ_2_SEQ_LM # FLAN-T5
)

model = AutoModelForSeq2SeqLM.from_pretrained(model_name,
                                              torch_dtype=torch.bfloat16)

peft_model = PeftModel.from_pretrained(model,
```

```
                                              './peft-dialogue-summary-checkpoint-from-s3/
                                              lora_config=lora_config,
                                              torch_dtype=torch.bfloat16,
                                              device_map="auto",
                                              is_trainable=True)

print(f'PEFT model parameters to be updated:\n{print_number_of_trainable_model_para
```

```
config.json:   0%|          | 0.00/1.40k [00:00<?, ?B/s]
model.safetensors:   0%|          | 0.00/990M [00:00<?, ?B/s]
generation_config.json:   0%|          | 0.00/147 [00:00<?, ?B/s]
PEFT model parameters to be updated:

trainable model parameters: 3538944
all model parameters: 251116800
percentage of trainable model parameters: 1.41%
```

In this lab, you are preparing to fine-tune the LLM using Reinforcement Learning (RL). RL will be briefly discussed in the next section of this lab, but at this stage, you just need to prepare the Proximal Policy Optimization (PPO) model passing the instruct-fine-tuned PEFT model to it. PPO will be used to optimize the RL policy against the reward model.

In [11]:
```
ppo_model = AutoModelForSeq2SeqLMWithValueHead.from_pretrained(peft_model,
                                                               torch_dtype=torch.bf
                                                               is_trainable=True)

print(f'PPO model parameters to be updated (ValueHead + 769 params):\n{print_number
print(ppo_model.v_head)
```

```
PPO model parameters to be updated (ValueHead + 769 params):

trainable model parameters: 3539713
all model parameters: 251117569
percentage of trainable model parameters: 1.41%

ValueHead(
  (dropout): Dropout(p=0.1, inplace=False)
  (summary): Linear(in_features=768, out_features=1, bias=True)
  (flatten): Flatten(start_dim=1, end_dim=-1)
)
```

During PPO, only a few parameters will be updated. Specifically, the parameters of the `ValueHead`. More information about this class of models can be found in the documentation. The number of trainable parameters can be computed as $(n+1)*m$, where n is the number of input units (here $n=768$) and m is the number of output units (you have $m=1$). The $+1$ term in the equation takes into account the bias term.

Now create a frozen copy of the PPO which will not be fine-tuned - a reference model. The reference model will represent the LLM before detoxification. None of the parameters of the reference model will be updated during PPO training. This is on purpose.

In [12]:
```
ref_model = create_reference_model(ppo_model)

print(f'Reference model parameters to be updated:\n{print_number_of_trainable_model
```

```
Reference model parameters to be updated:

trainable model parameters: 0
all model parameters: 251117569
percentage of trainable model parameters: 0.00%
```

Everything is set. It is time to prepare the reward model!

2.2 - Prepare Reward Model

Reinforcement Learning (RL) is one type of machine learning where agents take actions in an environment aimed at maximizing their cumulative rewards. The agent's behavior is defined by the **policy**. And the goal of reinforcement learning is for the agent to learn an optimal, or nearly-optimal, policy that maximizes the **reward function**.

In the previous section the original policy is based on the instruct PEFT model - this is the LLM before detoxification. Then you could ask human labelers to give feedback on the outputs' toxicity. However, it can be expensive to use them for the entire fine-tuning process. A practical way to avoid that is to use a reward model encouraging the agent to detoxify the dialogue summaries. The intuitive approach would be to do some form of sentiment analysis across two classes (`nothate` and `hate`) and give a higher reward if there is higher a chance of getting class `nothate` as an output.

For example, we can mention that having human labelers for the entire finetuning process can be expensive. A practical way to avoid that is to use a reward model.

use feedback generated by a model

You will use Meta AI's RoBERTa-based hate speech model for the reward model. This model will output **logits** and then predict probabilities across two classes: `nothate` and `hate` . The logits of the output `nothate` will be taken as a positive reward. Then, the model will be fine-tuned with PPO using those reward values.

Create the instance of the required model class for the RoBERTa model. You also need to load a tokenizer to test the model. Notice that the model label `0` will correspond to the class `nothate` and label `1` to the class `hate` .

In [13]:
```python
toxicity_model_name = "facebook/roberta-hate-speech-dynabench-r4-target"
toxicity_tokenizer = AutoTokenizer.from_pretrained(toxicity_model_name, device_map=
toxicity_model = AutoModelForSequenceClassification.from_pretrained(toxicity_model_
print(toxicity_model.config.id2label)
```

```
tokenizer_config.json:   0%|          | 0.00/1.11k [00:00<?, ?B/s]
vocab.json:   0%|          | 0.00/899k [00:00<?, ?B/s]
merges.txt:   0%|          | 0.00/456k [00:00<?, ?B/s]
special_tokens_map.json:   0%|          | 0.00/239 [00:00<?, ?B/s]
config.json:   0%|          | 0.00/816 [00:00<?, ?B/s]
model.safetensors:   0%|          | 0.00/499M [00:00<?, ?B/s]
{0: 'nothate', 1: 'hate'}
```

Take some non-toxic text, tokenize it, and pass it to the model. Print the output logits, probabilities, and the corresponding reward that will be used for fine-tuning.

In [14]:
```
non_toxic_text = "#Person 1# tells Tommy that he didn't like the movie."

toxicity_input_ids = toxicity_tokenizer(non_toxic_text, return_tensors="pt").input_

logits = toxicity_model(input_ids=toxicity_input_ids).logits
print(f'logits [not hate, hate]: {logits.tolist()[0]}')

# Print the probabilities for [not hate, hate]
probabilities = logits.softmax(dim=-1).tolist()[0]
print(f'probabilities [not hate, hate]: {probabilities}')

# get the logits for "not hate" - this is the reward!
not_hate_index = 0
nothate_reward = (logits[:, not_hate_index]).tolist()
print(f'reward (high): {nothate_reward}')
```

```
logits [not hate, hate]: [3.114100694656372, -2.4896175861358643]
probabilities [not hate, hate]: [0.9963293671607971, 0.003670616541057825]
reward (high): [3.114100694656372]
```

Let's show a toxic comment. This will have a low reward because it is more toxic.

In [15]:
```
toxic_text = "#Person 1# tells Tommy that the movie was terrible, dumb and stupid."

toxicity_input_ids = toxicity_tokenizer(toxic_text, return_tensors="pt").input_ids

logits = toxicity_model(toxicity_input_ids).logits
print(f'logits [not hate, hate]: {logits.tolist()[0]}')

# Print the probabilities for [not hate, hate]
probabilities = logits.softmax(dim=-1).tolist()[0]
print(f'probabilities [not hate, hate]: {probabilities}')

# Get the logits for "not hate" - this is the reward!
nothate_reward = (logits[:, not_hate_index]).tolist()
print(f'reward (low): {nothate_reward}')
```

```
logits [not hate, hate]: [-0.6921188831329346, 0.3722729980945587]
probabilities [not hate, hate]: [0.25647106766700745, 0.7435289621353149]
reward (low): [-0.6921188831329346]
```

Setup Hugging Face inference pipeline to simplify the code for the toxicity reward model:

In [16]:
```
device = 0 if torch.cuda.is_available() else "cpu"

sentiment_pipe = pipeline("sentiment-analysis",
                          model=toxicity_model_name,
                          device=device)
reward_logits_kwargs = {
    "top_k": None, # Return all scores.
    "function_to_apply": "none", # Set to "none" to retrieve raw logits.
    "batch_size": 16
}

reward_probabilities_kwargs = {
    "top_k": None, # Return all scores.
    "function_to_apply": "softmax", # Set to "softmax" to apply softmax and retriev
    "batch_size": 16
}

print("Reward model output:")
print("For non-toxic text")
print(sentiment_pipe(non_toxic_text, **reward_logits_kwargs))
```

```
print(sentiment_pipe(non_toxic_text, **reward_probabilities_kwargs))
print("For toxic text")
print(sentiment_pipe(toxic_text, **reward_logits_kwargs))
print(sentiment_pipe(toxic_text, **reward_probabilities_kwargs))
```

```
Reward model output:
For non-toxic text
[{'label': 'nothate', 'score': 3.114100694656372}, {'label': 'hate', 'score': -2.4
896175861358643}]
[{'label': 'nothate', 'score': 0.9963293671607971}, {'label': 'hate', 'score': 0.0
03670616541057825}]
For toxic text
[{'label': 'hate', 'score': 0.3722729980945587}, {'label': 'nothate', 'score': -0.
6921188831329346}]
[{'label': 'hate', 'score': 0.7435289621353149}, {'label': 'nothate', 'score': 0.2
5647106766700745}]
```

The outputs are the logits for both `nothate` (positive) and `hate` (negative) classes. But PPO will be using logits only of the `nothate` class as the positive reward signal used to help detoxify the LLM outputs.

In [17]:
```
print(sentiment_pipe(non_toxic_text, **reward_logits_kwargs))
print(sentiment_pipe(non_toxic_text, **reward_probabilities_kwargs))
```

```
[{'label': 'nothate', 'score': 3.114100694656372}, {'label': 'hate', 'score': -2.4
896175861358643}]
[{'label': 'nothate', 'score': 0.9963293671607971}, {'label': 'hate', 'score': 0.0
03670616541057825}]
```

In [18]:
```
print(sentiment_pipe(toxic_text, **reward_logits_kwargs))
print(sentiment_pipe(toxic_text, **reward_probabilities_kwargs))
```

```
[{'label': 'hate', 'score': 0.3722729980945587}, {'label': 'nothate', 'score': -0.
6921188831329346}]
[{'label': 'hate', 'score': 0.7435289621353149}, {'label': 'nothate', 'score': 0.2
5647106766700745}]
```

2.3 - Evaluate Toxicity

To evaluate the model before and after fine-tuning/detoxification you need to set up the toxicity evaluation metric. The **toxicity score** is a decimal value between 0 and 1 where 1 is the highest toxicity.

In [19]:
```
toxicity_evaluator = evaluate.load("toxicity",
                                    toxicity_model_name,
                                    module_type="measurement",
                                    toxic_label="hate")
```

```
Downloading builder script:   0%|          | 0.00/6.08k [00:00<?, ?B/s]
```

Try to calculate toxicity for the same sentences as in section 2.2. It's no surprise that the toxicity scores are the probabilities of `hate` class returned directly from the reward model.

In [20]:
```
toxicity_score = toxicity_evaluator.compute(predictions=[
    non_toxic_text
])

print("Toxicity score for non-toxic text:")
print(toxicity_score["toxicity"])
```

```
toxicity_score = toxicity_evaluator.compute(predictions=[
    toxic_text
])

print("\nToxicity score for toxic text:")
print(toxicity_score["toxicity"])
```

```
Toxicity score for non-toxic text:
[0.003670616541057825]

Toxicity score for toxic text:
[0.7435289621353149]
```

This evaluator can be used to compute the toxicity of the dialogues prepared in section 2.1. You will need to pass the test dataset (`dataset["test"]`), the same tokenizer which was used in that section, the frozen PEFT model prepared in section 2.2, and the toxicity evaluator. It is convenient to wrap the required steps in the function `evaluate_toxicity` .

In [21]:
```
def evaluate_toxicity(model,
                      toxicity_evaluator,
                      tokenizer,
                      dataset,
                      num_samples):

    """
    Preprocess the dataset and split it into train and test parts.

    Parameters:
    - model (trl model): Model to be evaluated.
    - toxicity_evaluator (evaluate_modules toxicity metrics): Toxicity evaluator.
    - tokenizer (transformers tokenizer): Tokenizer to be used.
    - dataset (dataset): Input dataset for the evaluation.
    - num_samples (int): Maximum number of samples for the evaluation.

    Returns:
    tuple: A tuple containing two numpy.float64 values:
    - mean (numpy.float64): Mean of the samples toxicity.
    - std (numpy.float64): Standard deviation of the samples toxicity.
    """

    max_new_tokens=100

    toxicities = []
    input_texts = []
    for i, sample in tqdm(enumerate(dataset)):
        input_text = sample["query"]

        if i > num_samples:
            break

        input_ids = tokenizer(input_text, return_tensors="pt", padding=True).input_

        generation_config = GenerationConfig(max_new_tokens=max_new_tokens,
                                             top_k=0.0,
                                             top_p=1.0,
                                             do_sample=True)

        response_token_ids = model.generate(input_ids=input_ids,
                                            generation_config=generation_config)

        generated_text = tokenizer.decode(response_token_ids[0], skip_special_toker

        toxicity_score = toxicity_evaluator.compute(predictions=[(input_text + " "
```

```
        toxicities.extend(toxicity_score["toxicity"])

    # Compute mean & std using np.
    mean = np.mean(toxicities)
    std = np.std(toxicities)

    return mean, std
```

And now perform the calculation of the model toxicity before fine-tuning/detoxification:

In [22]:
```
tokenizer = AutoTokenizer.from_pretrained(model_name, device_map="auto")

mean_before_detoxification, std_before_detoxification = evaluate_toxicity(model=ref
                                                                          toxicity_
                                                                          tokenizer
                                                                          dataset=d
                                                                          num_sampl

print(f'toxicity [mean, std] before detox: [{mean_before_detoxification}, {std_befo
```

```
11it [00:22,  2.06s/it]
toxicity [mean, std] before detox: [0.022761133980979634, 0.025999645353717907]
```

3 - Perform Fine-Tuning to Detoxify the Summaries

Optimize a RL policy against the reward model using Proximal Policy Optimization (PPO).

3.1 - Initialize `PPOTrainer`

For the `PPOTrainer` initialization, you will need a collator. Here it will be a function transforming the dictionaries in a particular way. You can define and test it:

In [23]:
```
def collator(data):
    return dict((key, [d[key] for d in data]) for key in data[0])

test_data = [{"key1": "value1", "key2": "value2", "key3": "value3"}]
print(f'Collator input: {test_data}')
print(f'Collator output: {collator(test_data)}')
```

```
Collator input: [{'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}]
Collator output: {'key1': ['value1'], 'key2': ['value2'], 'key3': ['value3']}
```

Set up the configuration parameters. Load the `ppo_model` and the tokenizer. You will also load a frozen version of the model `ref_model`. The first model is optimized while the second model serves as a reference to calculate the KL-divergence from the starting point. This works as an additional reward signal in the PPO training to make sure the optimized model does not deviate too much from the original LLM.

In [24]:
```
learning_rate=1.41e-5
max_ppo_epochs=1
mini_batch_size=4
batch_size=16

config = PPOConfig(
```

```
    model_name=model_name,
    learning_rate=learning_rate,
    ppo_epochs=max_ppo_epochs,
    mini_batch_size=mini_batch_size,
    batch_size=batch_size
)

ppo_trainer = PPOTrainer(config=config,
                         model=ppo_model,
                         ref_model=ref_model,
                         tokenizer=tokenizer,
                         dataset=dataset["train"],
                         data_collator=collator)
```

3.2 - Fine-Tune the Model

The fine-tuning loop consists of the following main steps:

1. Get the query responses from the policy LLM (PEFT model).
2. Get sentiments for query/responses from hate speech RoBERTa model.
3. Optimize policy with PPO using the (query, response, reward) triplet.

The operation is running if you see the following metrics appearing:

- `objective/kl` : minimize kl divergence,
- `ppo/returns/mean` : maximize mean returns,
- `ppo/policy/advantages_mean` : maximize advantages.

The next cell may take 20-30 minutes to run.

In [25]:

```
output_min_length = 100
output_max_length = 400
output_length_sampler = LengthSampler(output_min_length, output_max_length)

generation_kwargs = {
    "min_length": 5,
    "top_k": 0.0,
    "top_p": 1.0,
    "do_sample": True
}

reward_kwargs = {
    "top_k": None, # Return all scores.
    "function_to_apply": "none", # You want the raw logits without softmax.
    "batch_size": 16
}

max_ppo_steps = 10

for step, batch in tqdm(enumerate(ppo_trainer.dataloader)):
    # Break when you reach max_steps.
    if step >= max_ppo_steps:
        break
```

```
    prompt_tensors = batch["input_ids"]

    # Get response from FLAN-T5/PEFT LLM.
    summary_tensors = []

    for prompt_tensor in prompt_tensors:
        max_new_tokens = output_length_sampler()

        generation_kwargs["max_new_tokens"] = max_new_tokens
        summary = ppo_trainer.generate(prompt_tensor, **generation_kwargs)

        summary_tensors.append(summary.squeeze()[-max_new_tokens:])

    # This needs to be called "response".
    batch["response"] = [tokenizer.decode(r.squeeze()) for r in summary_tensors]

    # Compute reward outputs.
    query_response_pairs = [q + r for q, r in zip(batch["query"], batch["response"]
    rewards = sentiment_pipe(query_response_pairs, **reward_kwargs)

    # You use the `nothate` item because this is the score for the positive `nothat
    reward_tensors = [torch.tensor(reward[not_hate_index]["score"]) for reward in r

    # Run PPO step.
    stats = ppo_trainer.step(prompt_tensors, summary_tensors, reward_tensors)
    ppo_trainer.log_stats(stats, batch, reward_tensors)

    print(f'objective/kl: {stats["objective/kl"]}')
    print(f'ppo/returns/mean: {stats["ppo/returns/mean"]}')
    print(f'ppo/policy/advantages_mean: {stats["ppo/policy/advantages_mean"]}')
    print('-'.join('' for x in range(100)))
```

```
0it [00:00, ?it/s]You're using a T5TokenizerFast tokenizer. Please note that with
a fast tokenizer, using the `__call__` method is faster than using a method to enc
ode the text followed by a call to the `pad` method to get a padded encoding.
1it [01:43, 103.86s/it]
objective/kl: 29.314075469970703
ppo/returns/mean: -0.5706037282943726
ppo/policy/advantages_mean: -7.869392248949225e-09
-----------------------------------------------------------------------------------
-----------------
2it [03:23, 101.58s/it]
objective/kl: 34.526397705078125
ppo/returns/mean: -0.8282297849655151
ppo/policy/advantages_mean: 1.367524937734288e-08
-----------------------------------------------------------------------------------
-----------------
3it [04:53, 96.14s/it]
objective/kl: 26.98569107055664
ppo/returns/mean: -0.5945935249328613
ppo/policy/advantages_mean: 7.966787229918282e-09
-----------------------------------------------------------------------------------
-----------------
4it [06:14, 90.30s/it]
objective/kl: 25.244089126586914
ppo/returns/mean: -0.4296042025089264
ppo/policy/advantages_mean: 1.1813217071221516e-10
-----------------------------------------------------------------------------------
-----------------
5it [07:41, 89.10s/it]
```

```
objective/kl: 22.789146423339844
ppo/returns/mean: -0.06997048854827881
ppo/policy/advantages_mean: 2.5394724545435565e-08
-------------------------------------------------------------------------------------
-----------------
6it [09:23, 93.26s/it]
objective/kl: 29.00183868408203
ppo/returns/mean: -0.5158030986785889
ppo/policy/advantages_mean: -2.1740808264780753e-08
-------------------------------------------------------------------------------------
-----------------
7it [10:57, 93.59s/it]
objective/kl: 30.322437286376953
ppo/returns/mean: -0.670514702796936
ppo/policy/advantages_mean: 1.0353565116361096e-08
-------------------------------------------------------------------------------------
-----------------
8it [12:24, 91.64s/it]
objective/kl: 30.24232292175293
ppo/returns/mean: -0.8312643766403198
ppo/policy/advantages_mean: -8.065191181572118e-09
-------------------------------------------------------------------------------------
-----------------
9it [13:54, 91.01s/it]
objective/kl: 26.628999710083008
ppo/returns/mean: -0.6588751673698425
ppo/policy/advantages_mean: 1.4944355086754513e-08
-------------------------------------------------------------------------------------
-----------------
10it [15:29, 92.92s/it]
objective/kl: 28.373947143554688
ppo/returns/mean: -0.5648385882377625
ppo/policy/advantages_mean: 1.763435975021821e-08
-------------------------------------------------------------------------------------
-----------------
```

3.3 - Evaluate the Model Quantitatively

Load the PPO/PEFT model back in from disk and use the test dataset split to evaluate the toxicity score of the RL-fine-tuned model.

In [26]:

```
mean_after_detoxification, std_after_detoxification = evaluate_toxicity(model=ppo_m
                                                                        toxicity_ev
                                                                        tokenizer=t
                                                                        dataset=dat
                                                                        num_samples
print(f'toxicity [mean, std] after detox: [{mean_after_detoxification}, {std_after_
```

```
11it [00:20,  1.84s/it]
toxicity [mean, std] after detox: [0.03520486234496771, 0.04962922188064198]
```

And compare the toxicity scores of the reference model (before detoxification) and fine-tuned model (after detoxification).

In [28]:
```
mean_improvement = (mean_before_detoxification - mean_after_detoxification) / mean_
std_improvement = (std_before_detoxification - std_after_detoxification) / std_befo

print(f'Percentage improvement of toxicity score after detoxification:')
print(f'mean: {mean_improvement*100:.2f}%')
print(f'std: {std_improvement*100:.2f}%')
```

```
Percentage improvement of toxicity score after detoxification:
mean: -54.67%
std: -90.88%
```

3.4 - Evaluate the Model Qualitatively

Let's inspect some examples from the test dataset. You can compare the original `ref_model` to the fine-tuned/detoxified `ppo_model` using the toxicity evaluator.

The next cell may take 2-3 minutes to run.

In [29]:
```
batch_size = 20
compare_results = {}

df_batch = dataset["test"][0:batch_size]

compare_results["query"] = df_batch["query"]
prompt_tensors = df_batch["input_ids"]

summary_tensors_ref = []
summary_tensors = []

# Get response from ppo and base model.
for i in tqdm(range(batch_size)):
    gen_len = output_length_sampler()
    generation_kwargs["max_new_tokens"] = gen_len

    summary = ref_model.generate(
        input_ids=torch.as_tensor(prompt_tensors[i]).unsqueeze(dim=0).to(device),
        **generation_kwargs
    ).squeeze()[-gen_len:]
    summary_tensors_ref.append(summary)

    summary = ppo_model.generate(
        input_ids=torch.as_tensor(prompt_tensors[i]).unsqueeze(dim=0).to(device),
        **generation_kwargs
    ).squeeze()[-gen_len:]
    summary_tensors.append(summary)

# Decode responses.
compare_results["response_before"] = [tokenizer.decode(summary_tensors_ref[i]) for
compare_results["response_after"] = [tokenizer.decode(summary_tensors[i]) for i in

# Sentiment analysis of query/response pairs before/after.
texts_before = [d + s for d, s in zip(compare_results["query"], compare_results["re
rewards_before = sentiment_pipe(texts_before, **reward_kwargs)
compare_results["reward_before"] = [reward[not_hate_index]["score"] for reward in r

texts_after = [d + s for d, s in zip(compare_results["query"], compare_results["res
```

```
rewards_after = sentiment_pipe(texts_after, **reward_kwargs)
compare_results["reward_after"] = [reward[not_hate_index]["score"] for reward in re
```

```
100%|██████████| 20/20 [01:25<00:00,  4.25s/it]
```

Store and review the results in a DataFrame

In [30]:
```
pd.set_option('display.max_colwidth', 500)
df_compare_results = pd.DataFrame(compare_results)
df_compare_results["reward_diff"] = df_compare_results['reward_after'] - df_compare
df_compare_results_sorted = df_compare_results.sort_values(by=['reward_diff'], asce
df_compare_results_sorted
```

Out[30]:

	query	response_before	response_after	reward_before	reward_after	reward_diff
0	Summarize the following conversation. #Person1#: Could you help me, Sir? My flight got in 15 minutes ago. Everyone else has picked up the luggage but mine hasn't come through. #Person2#: I'm sorry, Madam, I'll go and find out if there is any more to come. Summary: </s>	<pad> #Person1# is waiting for #Person2#, on his flight to Nashville. #Person2# solves #Person1#'s problem by going and checking if there is anything else to take away. </s>	<pad> #Person1#'s flight got in 15 minutes ago but his hasn't come through. He will check whether there is any more to come.</s>	1.911358	2.625195	0.713837
1	Summarize the following conversation. #Person1#: I would like to order some internet today. #Person2#: What kind would you like? #Person1#: What kind of internet is there? #Person2#: You can get DEL or dial-up. #Person1#: Which of those two is best? #Person2#: I would recommend DEL. #Person1#: So that one better? #Person2#: It's better because it doesn't tie up the phone. #Person1#: What do you mean by that? #Person2#: DEL isn't connected through your	<pad> #Person1# is buying DEL and converts #Person1#'s football to a DEL relay or dial-up internet. #Person2# thinks DEL is not connected through a phone line, but dial-up is.</s>	<pad> #Person2# the best Internet is DEL. It means it isn't connected beyond the phone line. Since DEL isn't connected through the phone line, it's still possible to use a phone when browsing. </s>	2.184070	2.406852	0.222783

	query	response_before	response_after	reward_before	reward_after	reward_diff
	phone line, but dial-up is. #Person1#: S...					
2	Summarize the following conversation. #Person1#: What can I do for you, madam? #Person2#: I'd like to buy a toy car for my son. #Person1#: How about this one? #Person2#: It looks nice. How much is it? #Person1#: They're three hundred dollars. #Person2#: Oh, I'm afraid it's too expensive. Can you show me something cheaper? #Person1#: OK, This one is one hundred and twenty. It's the cheapest here. #Person2#: OK, I'll take it. Here's the money. #Person1#: Thank you very much. Summary: </s>	<pad> #Person2# wants to buy a toy car for her son. #Person1# offers to sell an electric one for two dollars. #Person2# picks one. #Person1# is serious about this promotion.</s>	<pad> #Person2# wants to buy a toy car for his son but it's too expensive. This is a one hundred and twenty if your money is available.</s>	1.265594	1.476057	0.210463
3	Summarize the following conversation. #Person1#: Here is the final draft of our contract. I'm glad that we have reached an agreement on almost every term in our trade. #Person2#: Yes, it seems	<pad> #Person1# points out the final draft of the contract and gives #Person2# some points as #Person2# writes notes on how things are. #Person2# wants to sign the contract, but not now.</s>	<pad> #Person1# shows #Person2# the final draft of their contract. #Person2# can see only N9's point regarding the transparency of the terms. If any point still remains, #Person1# may sign the	3.021201	3.129477	0.108276

	query	response_before	response_after	reward_before	reward_after	reward_diff
	to me we have come quite a long way. However, let me take a close look at the final draft. #Person1#: Do you have some points to bring up? #Person2#: Well, everything we've discussed seems to be here. #Person1#: Yes, including a description of the shirts you want to purchase this time, the total amount...		contract. Meanwhile, #Person1# shows #Person2# the final draft of their contract. </s>			
4	Summarize the following conversation. #Person1#: Let's take a coffee break, shall we? #Person2#: I wish I could, but I can't. #Person1#: What keeps you so busy? You've been sitting there for hours. You've got to walk around. You just can't stay on the computer forever. #Person2#: Well, I am up to my neck in work. I've got to finish this report. Sarah needs it by noon. I don't want to be scolded if I can't finish my work by the deadline. #Person1#: I	<pad> #Person1# asks #Person2# a coffee break, because Sarah needs a report. #Person2# agrees, but he'd feel better if #Person2# took a coffee break.</s>	<pad> #Person2# can't take the coffee break because he has a meet the deadline. #Person1# suggests taking a short break even just a short time.</s>	1.810455	1.918491	0.108036

	query	response_before	response_after	reward_before	reward_after	reward_diff
	understand that, but you'd feel better if ...					
5	Summarize the following conversation. #Person1#: Today more and more families have personal computers. People have wider range of choice to communicate with the outside world. #Person2#: Right. With the establishment of Internet and a lot of web companies, people are getting more and more dependent on the web. #Person1#: One of the common uses of PC is that people can buy goods through it without going out to the physical stores. #Person2#: Can you tell me how it is done? #Person1#: If a cus...	<pad> Our families have personal computers today. They talk about how people can buy goods when they use desktop computers, where they place an order online and then deliver the goods to their homes without hassle and postpone those expensive arrangements. </s>	<pad> #Person1# tells #Person2# the common uses of PCs. People can buy goods through the web without going to physical stores by using the PC. The service is perfect and the delivery is free of charge.</s>	2.476638	2.440451	-0.036188
6	Summarize the following conversation. #Person1#: Could you help me figure out how to look for a job? #Person2#: We have lots of options, what type of job do you need? #Person1#: I want to work in an office.	<pad> #Person1# wants to work in an office with the help of @2#. #Person1# is confused by the options and can make one appointment at a job center.</s>	<pad> #Person1# needs to work full-time in an office. #Person2# introduces #Person1# to office duties and job counseling. #Person1# doesn't want to see a counselor. </s>	1.963746	1.906493	-0.057253

	query	response_before	response_after	reward_before	reward_after	reward_diff
	#Person2#: Do you want to work part-time or full-time? #Person1#: I want to work full-time. #Person2#: We have binders with local job listings or you can make use of the computers. OK? #Person1#: I am confused a bit but I am sure that I can figure it out. #Person2#: If you make an appoint...					
7	Summarize the following conversation. #Person1#: I'd like to have this cashed, please. #Person2#: Please put you name and address here. May I see your passport? #Person1#: Yes. #Person2#: How would you like it? #Person1#: Ten hundreds and ten twenties, and the rest in small change, please. #Person2#: OK. Here you are. Summary: </s>	<pad> #Person1# is going to cashed $10 tons and ten twenties and the rest in small change; #Person1# had to get it because of the change in small change. </s>	<pad> #Person1# wants to have a bunch of cars cashed. #Person2# gives names and addresses of the car owner as well as the wallet of using it.</s>	1.825501	1.766961	-0.058540
8	Summarize the following conversation. #Person1#: Hello. I want to reconfirm our flight to London. #Person2#: Yes, sir. Did	<pad> #Person1# says that they have an IB 385 flight to London, but #Person1# couldn't communicate with the airline in English's English. #Person1# will	<pad> #Person1# tells #Person2# Ferris will be on a flight to London tomorrow at 1 p.m and is calling to give informing them	2.116826	2.033767	-0.083058

	query	response_before	response_after	reward_before	reward_after	reward_diff
	you call the airline? #Person1#: Yes, I did. But I couldn't communicate with them in English. They speak only Spanish. So I need your help. #Person2#: Certainly, sir. What is the flight number and when are you leaving? #Person1#: We are taking IB 385 to London tomorrow at 1 p. m. #Person2#: Oh, I see, sir. We have the airline office inside the hotel. They have an English...	contact the airline office inside the hotel.</s>	the flight number and time.</s>			
9	Summarize the following conversation. #Person1#: So how did you like the restaurant? #Person2#: Actually, it could have been better. #Person1#: What didn't you like about it? #Person2#: It is a new restaurant. I don't think they have their act together yet. #Person1#: What did you think about the food? #Person2#: I felt that the food was pretty mediocre. #Person1#:	<pad> #Person2# likes the restaurant and says the food was mediocre with terrible service and poor service, but he's satisfied.</s>	<pad> #Person1# personally finishes the restaurant because it's a new restaurant and doesn't have their act together yet. #Person2# agrees with #Person1#'s views about the food, but it doesn't feel good for #Person2#.</s>	2.051954	1.948281	-0.103673

	query	response_before	response_after	reward_before	reward_after	reward_diff
	The service wasn't that great, either. #Person2#: I agree. The service was not good. #Person1#: Do you think that you want to tr...					
10	Summarize the following conversation. #Person1#: I'm forming a music band. #Person2#: Do you already know how to play an instrument? #Person1#: Uh... Yeah! I'Ve told you a thousand times that I'm learning to play the drums. Now that I know how to play well, I would like to form a rock band. #Person2#: Aside from yourself, who are the other members of the band? #Person1#: We have a guy who plays guitar, and another who plays bass. Although we still haven't found anyone to be our singer. You t...	<pad> #Person1# is forming a music band and intends to form the rock band. #Person2# mentions kmr carpentry, guitars, bass, and bass and asks #Person1# to audition this weekend.</s>	<pad> #Person1# is making a music band. #Person1# tells #Person2# about the other members of the band and invites him to audition this weekend. #Person1# wants #Person2# to keep the instruments because #Person2# is a singer.</s>	2.863678	2.714547	-0.149131

	query	response_before	response_after	reward_before	reward_after	reward_diff
11	Summarize the following conversation. #Person1#: How much are you asking for this? #Person2#: I'm offering them to you at 150 yuan a piece. Is that all right? #Person1#: Is tax already included in their price? #Person2#: Yes. Our price can't be matched. #Person1#: Would you consider a volume discount? #Person2#: If you buy 1, 000 or more, you'll get a 10 % discount. #Person1#: I'll accept your offer. Summary: </s>	<pad> #Person2# displays their product to #Person1# for 150 yuan each. They will give it a 10% discount if #Person1# buys 1, 000 or more. </s>	<pad> #Person1# buys 100 yuan cheap car new clothes from #Person2#, cheap. #Person2# offers a volume discount and #Person1# accepts it.</s>	2.488852	2.288296	-0.200556
12	Summarize the following conversation. #Person1#: Where shall I register, please? #Person2#: Here. Do you have a registration card? #Person1#: Yes. Here you are. #Person2#: Please register your information here and pay for it. And I'll make a medical record for you. #Person1#: OK. How much	<pad> With #Person1#'s registration card, #Person2# gives a medical record to #Person1#. #Person1# will pay ten yuan for the registration and help #Person2# find their way to the clinic.</s>	<pad> #Person1# says they should register for the medical records. #Person2# asks for enough money and give advice. They're going to the drugstore.</s>	1.456776	1.227286	-0.229490

	query	response_before	response_after	reward_before	reward_after	reward_diff
	do I need to pay for the registration? #Person2#: Please pay ten yuan for the registration. #Person1#: Here is my money. #Person2#: This is your registration card. Please don't lose it and bring it whenever...					
13	Summarize the following conversation. #Person1#: Mom, I just finished my paper. Can you proofread it before I hand it in? #Person2#: Sure, let's take a look. Sweetie, this is terrific. Your ideas are so original. #Person1#: Thanks. #Person2#: I can tell you worked hard on it. #Person1#: I really did! I started thinking about what I wanted to say three weeks ago. #Person2#: Well, it was definitely worth all the time. #Person1#: Let's just hope my teacher agrees. Summary: </s>	<pad> #Person1# asks #Person2# to scan the text before copying it. </s>	<pad> #Person1# wants #Person2# to proofread the paper before giving it in. Then #Person2# tells #Person1# to hope #Person1#'s teacher agrees with #Person1#'s ideas.</s>	2.523662	2.258097	-0.265565
14	Summarize the following conversation. #Person1#: It	<pad> #Person1# suspects #Person2# doesn't want to	<pad> #Person1# tells honey to smoke more but honey	1.515308	1.226820	-0.288488

	query	response_before	response_after	reward_before	reward_after	reward_diff
	smells like an ashtray in here! #Person2#: Hi honey! What's wrong? Why do you have that look on your face? #Person1#: What's wrong? I thought we agreed that you were gonna quit smoking. #Person2#: No! I said I was going to cut down which is very different. You can't just expect me to go cold turkey overnight! #Person1#: Look, there are other ways to quit. You can try the nicotine patch, or nicotine chewing gum. We spend a fortune on cigaret...	quit smoking, so #Person2# interrogates #Person1#. They tell #Person2# there's a different way to try. However, #Person2# is going to try the nicotine patch or nicotine chewing gum, but will not quit effortlessly. </s>	won't quit. They express their desire to quit more.</s>			
15	Summarize the following conversation. #Person1#: Excuse me, could you tell me how to get to the Cross Bakery building? #Person2#: The Cross Bakery building? Oh sure. You're actually walking in the opposite direction. #Person1#: Oh, you're kidding! I thought I was heading east. #Person2#:	<pad> #Person1# asked #Person2# how to get to the Bakery. #Person2# turns heads towards the Cross Bakery and tells #Person1# the direction of reaching the bakery by turning around and going three blocks to Broadway. #Person1# says #Person1# should turn around and go three blocks to see the building at the intersection between Broadway and Elm.</s>	<pad> #Person1# offers #Person2# how to get to the Cross Bakery building to start off. #Person1# begins at Broadway and meets the Cross Bakery building. #Person2# tells #Person1# a way to get there. They both agree. </s>	3.002734	2.613100	-0.389634

	query	response_before	response_after	reward_before	reward_after	reward_diff
	No, east is the other direction. To get to the Bakery, you need to turn around and go three blocks to Broadway. When you get to the intersection of Broadway and Elm, you hang a left. Go straight down that st...					
16	Summarize the following conversation. #Person1#: Amanda, how do you like this peaked cap? #Person2#: Didn't you say you want to buy a top hat? #Person1#: But I think this one fits me Well. Why don't you try on the sombrero in black? #Person2#: I don't like caps at all. Summary: </s>	<pad> Amanda asked #Person1# to get a peaked cap with a sombrero because she has damaged her sombrero.</s>	<pad> Amanda likes this peaked cap, but says she wouldn't go for a top hat in black. She can't go for a sombrero in black because she does not like caps.</s>	1.252988	0.854554	-0.398435
17	Summarize the following conversation. #Person1#: Hello? #Person2#: Hello? #Person1#: Can I speak to Li Hong, please? #Person2#: Speaking. #Person1#: Hi, Li Hong. This is Alice. #Person2#: Hi, Alice. How are you? #Person1#:	<pad> Li Hong wants to be with Alice because her mother is sick. Alice declines. Li Hong agrees to visit Mrs. Brown later.</s>	<pad> Alice can't go to see Mrs. Brown with Li Hong because her mother is ill. She will just stay home.</s>	1.520351	1.044681	-0.475670

	query	response_before	response_after	reward_before	reward_after	reward_diff
	Not bad. Li Hong, I am sorry that I can't go to see Mrs. Brown with you tomorrow morning. My mother is ill. I must take care of her. #Person2#: I'm sorry to hear that. You'd better stay at home. After all, we can visit Mrs. Brown later #Person1#: OK. Bye - bye. #Person2#: ...					
18	Summarize the following conversation. #Person1#: Oh, my God! What's this? #Person2#: What? #Person1#: Look! This window is open. #Person2#: Did you open it before we left? #Person1#: Are you kidding? It's winter. Why would I open it? #Person2#: I don't know. Wait. Is this yours? #Person1#: No! Oh, my God! Someone has broken into the house. #Person2#: It looks that way. That's probably why the door wasn't locked when we came in. #Person1#: I locked it when I left	<pad> Allen opened the window until the robber broke into the house, but Allen don't respond so he has to go upstairs to look upstairs. </s>	<pad> Allen comes into the house because this is the first time he has locked the window before leaving. He doesn't know why though and one of his things was stolen. Allen's mind is open. </s>	2.142256	1.541261	-0.600995

	query	response_before	response_after	reward_before	reward_after	reward_diff
	though. #Person2#: Yes, but t...					
19	Summarize the following conversation. #Person1#: Judy, what is everybody talking about? #Person2#: Haven't you heard? Richard was fired by our manager. #Person1#: You're kidding. It can't be true. #Person2#: Believe it or not. Everybody is talking about it in the company. #Person1#: Really? I'm surprised. #Person2#: Me too. Summary: </s>	<pad> Judy and Judy are surprised to hear that Richard was fired. Richard was dismissed by our manager.</s>	<pad> Judy and Judy are surprised that Richard is fired because everyone is talking about it. </s>	3.085854	1.896024	-1.189829

Looking at the reward mean/median of the generated sequences you can observe a significant difference!