```
!pip install torch==2.5.1 torchvision==0.20.1 torchaudio==2.5.1 --
index-url https://download.pytorch.org/whl/cu124
!pip install transformers datasets evaluate -q

import nltk
! pip install transformers
!pip install tree_sitter==0.2.0
nltk.download('punkt')
nltk.download('punkt_tab')
! pip install evaluate
! pip install sacrebleu
! pip install wandb
```

```
Looking in indexes: https://download.pytorch.org/whl/cu124
Collecting torch==2.5.1
  Downloading https://download.pytorch.org/whl/cu124/torch-
2.5.1%2Bcu124-cp311-cp311-linux_x86_64.whl (908.3 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 908.3/908.3 MB 1.1 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.3/7.3 MB 113.3 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.4/3.4 MB 95.0 MB/s eta
0:00:00
ent already satisfied: filelock in /usr/local/lib/python3.11/dist-
packages (from torch==2.5.1) (3.18.0)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.11/dist-packages (from torch==2.5.1) (4.13.1)
Requirement already satisfied: networkx in
/usr/local/lib/python3.11/dist-packages (from torch==2.5.1) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.11/dist-packages (from torch==2.5.1) (3.1.6)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.11/dist-packages (from torch==2.5.1) (2025.3.2)
Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch==2.5.1)
  Downloading
https://download.pytorch.org/whl/cu124/nvidia_cuda_nvrtc_cu12-
12.4.127-py3-none-manylinux2014_x86_64.whl (24.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 88.3 MB/s eta
0:00:00
e-cu12==12.4.127 (from torch==2.5.1)
  Downloading
https://download.pytorch.org/whl/cu124/nvidia_cuda_runtime_cu12-
12.4.127-py3-none-manylinux2014_x86_64.whl (883 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 50.8 MB/s eta
0:00:00
 torch==2.5.1)
  Downloading
https://download.pytorch.org/whl/cu124/nvidia_cuda_cupti_cu12-
12.4.127-py3-none-manylinux2014_x86_64.whl (13.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 109.5 MB/s eta
```

```
0:00:00
 torch==2.5.1)
  Downloading
https://download.pytorch.org/whl/cu124/nvidia_cudnn_cu12-9.1.0.70-py3-
none-manylinux2014_x86_64.whl (664.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 1.4 MB/s eta
0:00:00
 torch==2.5.1)
  Downloading
https://download.pytorch.org/whl/cu124/nvidia_cublas_cu12-12.4.5.8-
py3-none-manylinux2014_x86_64.whl (363.4 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 2.4 MB/s eta
0:00:00
 torch==2.5.1)
  Downloading
https://download.pytorch.org/whl/cu124/nvidia_cufft_cu12-11.2.1.3-py3-
none-manylinux2014_x86_64.whl (211.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 9.8 MB/s eta
0:00:00
 torch==2.5.1)
  Downloading
https://download.pytorch.org/whl/cu124/nvidia_curand_cu12-10.3.5.147-
py3-none-manylinux2014_x86_64.whl (56.3 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 36.0 MB/s eta
0:00:00
 torch==2.5.1)
  Downloading
https://download.pytorch.org/whl/cu124/nvidia_cusolver_cu12-11.6.1.9-
py3-none-manylinux2014_x86_64.whl (127.9 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 16.1 MB/s eta
0:00:00
 torch==2.5.1)
  Downloading
https://download.pytorch.org/whl/cu124/nvidia_cusparse_cu12-
12.3.1.170-py3-none-manylinux2014_x86_64.whl (207.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 2.7 MB/s eta
0:00:00
ent already satisfied: nvidia-nccl-cu12==2.21.5 in
/usr/local/lib/python3.11/dist-packages (from torch==2.5.1) (2.21.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in
/usr/local/lib/python3.11/dist-packages (from torch==2.5.1) (12.4.127)
Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch==2.5.1)
  Downloading
https://download.pytorch.org/whl/cu124/nvidia_nvjitlink_cu12-12.4.127-
py3-none-manylinux2014_x86_64.whl (21.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 94.1 MB/s eta
0:00:00
 torch==2.5.1)
  Downloading https://download.pytorch.org/whl/triton-3.1.0-cp311-
```

```
cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (209.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 209.5/209.5 MB 9.6 MB/s eta
0:00:00
ent already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.11/dist-packages (from torch==2.5.1) (1.13.1)
Requirement already satisfied: numpy in
/usr/local/lib/python3.11/dist-packages (from torchvision==0.20.1)
(2.0.2)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/usr/local/lib/python3.11/dist-packages (from torchvision==0.20.1)
(11.1.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.11/dist-packages (from sympy==1.13.1-
>torch==2.5.1) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.11/dist-packages (from jinja2->torch==2.5.1)
(3.0.2)
Installing collected packages: triton, nvidia-nvjitlink-cu12, nvidia-
curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-
nvrtc-cu12, nvidia-cuda-cupti-cu12, nvidia-cublas-cu12, nvidia-
cusparse-cu12, nvidia-cudnn-cu12, nvidia-cusolver-cu12, torch,
torchvision, torchaudio
  Attempting uninstall: triton
    Found existing installation: triton 3.2.0
    Uninstalling triton-3.2.0:
      Successfully uninstalled triton-3.2.0
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
    Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
    Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
```

```
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.5.3.2
    Uninstalling nvidia-cublas-cu12-12.5.3.2:
      Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.1.3
    Uninstalling nvidia-cusparse-cu12-12.5.1.3:
      Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.3.0.75
    Uninstalling nvidia-cudnn-cu12-9.3.0.75:
      Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
  Attempting uninstall: torch
    Found existing installation: torch 2.6.0+cu124
    Uninstalling torch-2.6.0+cu124:
      Successfully uninstalled torch-2.6.0+cu124
  Attempting uninstall: torchvision
    Found existing installation: torchvision 0.21.0+cu124
    Uninstalling torchvision-0.21.0+cu124:
      Successfully uninstalled torchvision-0.21.0+cu124
  Attempting uninstall: torchaudio
    Found existing installation: torchaudio 2.6.0+cu124
    Uninstalling torchaudio-2.6.0+cu124:
      Successfully uninstalled torchaudio-2.6.0+cu124
Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-
cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-runtime-
cu12-12.4.127 nvidia-cudnn-cu12-9.1.0.70 nvidia-cufft-cu12-11.2.1.3
nvidia-curand-cu12-10.3.5.147 nvidia-cusolver-cu12-11.6.1.9 nvidia-
cusparse-cu12-12.3.1.170 nvidia-nvjitlink-cu12-12.4.127 torch-
2.5.1+cu124 torchaudio-2.5.1+cu124 torchvision-0.20.1+cu124 triton-
3.1.0
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 491.2/491.2 kB 26.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 84.0/84.0 kB 7.6 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 10.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 183.9/183.9 kB 16.6 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 143.5/143.5 kB 12.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.8/194.8 kB 16.3 MB/s eta
0:00:00
ERROR: pip's dependency resolver does not currently take into account
```

```
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec
2024.12.0 which is incompatible.
Requirement already satisfied: transformers in
/usr/local/lib/python3.11/dist-packages (4.50.3)
Requirement already satisfied: filelock in
/usr/local/lib/python3.11/dist-packages (from transformers) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.26.0 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.30.1)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.11/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.11/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.11/dist-packages (from transformers)
(2024.11.6)
Requirement already satisfied: requests in
/usr/local/lib/python3.11/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.21.1)
Requirement already satisfied: safetensors>=0.4.3 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
Requirement already satisfied: tqdm>=4.27 in
/usr/local/lib/python3.11/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub<1.0,>=0.26.0->transformers) (2024.12.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub<1.0,>=0.26.0->transformers) (4.13.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(2025.1.31)
Collecting tree_sitter==0.2.0
  Downloading tree_sitter-0.2.0.tar.gz (110 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 110.4/110.4 kB 7.5 MB/s eta
0:00:00
```

```
etadata (setup.py) ... e=tree_sitter-0.2.0-cp311-cp311-
linux_x86_64.whl size=389536
sha256=cd7c2e6f8c920c04209b395e7a8232827ef2a66d44a3eb11f96a8f5292c1241
9
  Stored in directory:
/root/.cache/pip/wheels/d9/6e/e2/b0126ad4f531cf09749b69518118f0ebf7bf3
134ed91c71abb
Successfully built tree_sitter
Installing collected packages: tree_sitter
Successfully installed tree_sitter-0.2.0

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.

Requirement already satisfied: evaluate in
/usr/local/lib/python3.11/dist-packages (0.4.3)
Requirement already satisfied: datasets>=2.0.0 in
/usr/local/lib/python3.11/dist-packages (from evaluate) (3.5.0)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.11/dist-packages (from evaluate) (2.0.2)
Requirement already satisfied: dill in /usr/local/lib/python3.11/dist-
packages (from evaluate) (0.3.8)
Requirement already satisfied: pandas in
/usr/local/lib/python3.11/dist-packages (from evaluate) (2.2.2)
Requirement already satisfied: requests>=2.19.0 in
/usr/local/lib/python3.11/dist-packages (from evaluate) (2.32.3)
Requirement already satisfied: tqdm>=4.62.1 in
/usr/local/lib/python3.11/dist-packages (from evaluate) (4.67.1)
Requirement already satisfied: xxhash in
/usr/local/lib/python3.11/dist-packages (from evaluate) (3.5.0)
Requirement already satisfied: multiprocess in
/usr/local/lib/python3.11/dist-packages (from evaluate) (0.70.16)
Requirement already satisfied: fsspec>=2021.05.0 in
/usr/local/lib/python3.11/dist-packages (from fsspec[http]>=2021.05.0-
>evaluate) (2024.12.0)
Requirement already satisfied: huggingface-hub>=0.7.0 in
/usr/local/lib/python3.11/dist-packages (from evaluate) (0.30.1)
Requirement already satisfied: packaging in
/usr/local/lib/python3.11/dist-packages (from evaluate) (24.2)
Requirement already satisfied: filelock in
/usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0-
>evaluate) (3.18.0)
Requirement already satisfied: pyarrow>=15.0.0 in
/usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0-
>evaluate) (18.1.0)
Requirement already satisfied: aiohttp in
/usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0-
>evaluate) (3.11.15)
```

```
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0-
>evaluate) (6.0.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.7.0-
>evaluate) (4.13.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.19.0-
>evaluate) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.19.0-
>evaluate) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.19.0-
>evaluate) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.19.0-
>evaluate) (2025.1.31)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas->evaluate)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.11/dist-packages (from pandas->evaluate)
(2025.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.11/dist-packages (from pandas->evaluate)
(2025.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (6.3.2)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (0.3.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (1.18.3)
Requirement already satisfied: six>=1.5 in
```

```
/usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2-
>pandas->evaluate) (1.17.0)
Collecting sacrebleu
  Downloading sacrebleu-2.5.1-py3-none-any.whl.metadata (51 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 51.8/51.8 kB 4.4 MB/s eta
0:00:00
 sacrebleu)
  Downloading portalocker-3.1.1-py3-none-any.whl.metadata (8.6 kB)
Requirement already satisfied: regex in
/usr/local/lib/python3.11/dist-packages (from sacrebleu) (2024.11.6)
Requirement already satisfied: tabulate>=0.8.9 in
/usr/local/lib/python3.11/dist-packages (from sacrebleu) (0.9.0)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.11/dist-packages (from sacrebleu) (2.0.2)
Collecting colorama (from sacrebleu)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Requirement already satisfied: lxml in /usr/local/lib/python3.11/dist-
packages (from sacrebleu) (5.3.1)
Downloading sacrebleu-2.5.1-py3-none-any.whl (104 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 104.1/104.1 kB 9.3 MB/s eta
0:00:00
a-0.4.6-py2.py3-none-any.whl (25 kB)
Downloading portalocker-3.1.1-py3-none-any.whl (19 kB)
Installing collected packages: portalocker, colorama, sacrebleu
Successfully installed colorama-0.4.6 portalocker-3.1.1 sacrebleu-
2.5.1
Requirement already satisfied: wandb in
/usr/local/lib/python3.11/dist-packages (0.19.9)
Requirement already satisfied: click!=8.0.0,>=7.1 in
/usr/local/lib/python3.11/dist-packages (from wandb) (8.1.8)
Requirement already satisfied: docker-pycreds>=0.4.0 in
/usr/local/lib/python3.11/dist-packages (from wandb) (0.4.0)
Requirement already satisfied: gitpython!=3.1.29,>=1.0.0 in
/usr/local/lib/python3.11/dist-packages (from wandb) (3.1.44)
Requirement already satisfied: platformdirs in
/usr/local/lib/python3.11/dist-packages (from wandb) (4.3.7)
Requirement already satisfied: protobuf!=4.21.0,!=5.28.0,<6,>=3.19.0
in /usr/local/lib/python3.11/dist-packages (from wandb) (5.29.4)
Requirement already satisfied: psutil>=5.0.0 in
/usr/local/lib/python3.11/dist-packages (from wandb) (5.9.5)
Requirement already satisfied: pydantic<3 in
/usr/local/lib/python3.11/dist-packages (from wandb) (2.11.2)
Requirement already satisfied: pyyaml in
/usr/local/lib/python3.11/dist-packages (from wandb) (6.0.2)
Requirement already satisfied: requests<3,>=2.0.0 in
/usr/local/lib/python3.11/dist-packages (from wandb) (2.32.3)
Requirement already satisfied: sentry-sdk>=2.0.0 in
/usr/local/lib/python3.11/dist-packages (from wandb) (2.25.1)
Requirement already satisfied: setproctitle in
```

```
/usr/local/lib/python3.11/dist-packages (from wandb) (1.3.5)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.11/dist-packages (from wandb) (75.2.0)
Requirement already satisfied: typing-extensions<5,>=4.4 in
/usr/local/lib/python3.11/dist-packages (from wandb) (4.13.1)
Requirement already satisfied: six>=1.4.0 in
/usr/local/lib/python3.11/dist-packages (from docker-pycreds>=0.4.0-
>wandb) (1.17.0)
Requirement already satisfied: gitdb<5,>=4.0.1 in
/usr/local/lib/python3.11/dist-packages (from gitpython!
=3.1.29,>=1.0.0->wandb) (4.0.12)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.11/dist-packages (from pydantic<3->wandb)
(0.7.0)
Requirement already satisfied: pydantic-core==2.33.1 in
/usr/local/lib/python3.11/dist-packages (from pydantic<3->wandb)
(2.33.1)
Requirement already satisfied: typing-inspection>=0.4.0 in
/usr/local/lib/python3.11/dist-packages (from pydantic<3->wandb)
(0.4.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.0.0-
>wandb) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.0.0-
>wandb) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.0.0-
>wandb) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.0.0-
>wandb) (2025.1.31)
Requirement already satisfied: smmap<6,>=3.0.1 in
/usr/local/lib/python3.11/dist-packages (from gitdb<5,>=4.0.1-
>gitpython!=3.1.29,>=1.0.0->wandb) (5.0.2)

from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

# Define the model name for the CodeT5 small variant from Salesforce
model_name = "Salesforce/codet5-small"

# Load the pre-trained tokenizer
tokenizer = AutoTokenizer.from_pretrained (model_name)

# Load the pre-trained CodeT5 model
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)

print(f"Successfully loaded the model and tokenizer for
{model_name}.")
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/
_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your
settings tab (https://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to
access public models or datasets.
  warnings.warn(
```


```
{"model_id":"990380ce3aed4eefaaeea0a66e0561ad","version_major":2,"vers
ion_minor":0}

{"model_id":"14060c2126f04429a552b48b6507fa64","version_major":2,"vers
ion_minor":0}

{"model_id":"22f5e7af194948c4986ce1680678f6b5","version_major":2,"vers
ion_minor":0}

{"model_id":"17ccdb2546034b28beb489c42e066033","version_major":2,"vers
ion_minor":0}

{"model_id":"b55e4136a78a4beb96389c14fc7f2069","version_major":2,"vers
ion_minor":0}

{"model_id":"057e98e4b796443798a48ac9bb8755f5","version_major":2,"vers
ion_minor":0}

{"model_id":"ed462f555d07480aa665ec9180b6e641","version_major":2,"vers
ion_minor":0}
```


```
Successfully loaded the model and tokenizer for Salesforce/codet5-
small.
```

```python
import pandas as pd
import re

def mask_if_condition(function_code):
    """
    Flatten the function code by removing extra whitespace and
newlines,
    then mask the first if condition by replacing its condition with
<mask>.

    Args:
        function_code (str): The original function code as a string.

    Returns:
        str: The flattened function code with the if condition masked.
    """
    # Flatten the code by joining all lines with a single space
```

```
    flattened = " ".join(function_code.strip().split())
    # Use regex to find the first if condition and replace the
condition part
    # The pattern looks for an 'if' followed by one or more characters
until a colon
    # and replaces that with 'if <mask>:'
    masked = re.sub(r'\bif\s+([^:]+):', 'if <mask>:', flattened,
count=1)
    return masked

# --- Example Usage ---
# This example is based on the provided sample:
example_function = """def check_positive(num):
    if num > 0:
         return "Positive"
    else:
         return "Non-Positive"
"""

print("Example before masking:")
print(example_function)
print("\nExample after masking and flattening:")
print(mask_if_condition(example_function))

# --- Processing the Dataset ---
# Load your training dataset (adjust the file path as needed)
# Assume your CSV has at least the following columns:
# 'cleaned_method' (the full function code),
# 'target_block' (the if condition to be predicted),
# 'tokens_in_method' (other auxiliary tokens).
df_train = pd.read_csv("ft_train.csv")

# Apply the masking function on the 'cleaned_method' column
# to create a new column with the modified code.
df_train['masked_function'] =
df_train['cleaned_method'].apply(mask_if_condition)

# Inspect the transformed functions along with the target if block.
print("\nTransformed functions (sample):")
print(df_train[['cleaned_method', 'masked_function',
'target_block']].head())

# You can repeat the same process for the validation and test
datasets:
df_valid = pd.read_csv("ft_valid.csv")
df_valid['masked_function'] =
df_valid['cleaned_method'].apply(mask_if_condition)

df_test = pd.read_csv("ft_test.csv")
```

```
df_test['masked_function'] =
df_test['cleaned_method'].apply(mask_if_condition)

Example before masking:
def check_positive(num):
    if num > 0:
         return "Positive"
    else:
         return "Non-Positive"


Example after masking and flattening:
def check_positive(num): if <mask>: return "Positive" else: return
"Non-Positive"

Transformed functions (sample):
                                          cleaned_method  \
0  def _resolve_lib_imported_symbols(self, lib, i...
1  def make_docs_directory(output_dir, name):\n  ...
2  def assert_results(self, results, activities, ...
3  def for_file(cls, filename: str, modname: str)...
4  def merge_dicts(source: Dict, destination: Dic...

                                         masked_function  \
0  def _resolve_lib_imported_symbols(self, lib, i...
1  def make_docs_directory(output_dir, name): if ...
2  def assert_results(self, results, activities, ...
3  def for_file(cls, filename: str, modname: str)...
4  def merge_dicts(source: Dict, destination: Dic...

                                           target_block
0                                    if generic_refs :
1  if not isdir ( pjoin ( output_dir , name , str...
2           if hasattr ( result , "extra_context" ) :
3                if ".egg" + path . sep in filename :
4                    if isinstance ( value , dict ) :

{"model_id":"a0de327f9b774431b82ee79f1a403a56","version_major":2,"vers
ion_minor":0}

import torch
from torch.utils.data import Dataset, DataLoader

# Define a custom Dataset for fine-tuning CodeT5.
class FineTuneDataset(Dataset):
    def __init__(self, dataframe, tokenizer, max_input_length=512,
max_target_length=128):
        """
        Args:
            dataframe (pd.DataFrame): DataFrame with columns
'masked_function' and 'target_block'.
```

```
            tokenizer (PreTrainedTokenizer): Tokenizer loaded from
Hugging Face.
            max_input_length (int): Maximum length for the input
sequence.
            max_target_length (int): Maximum length for the target
sequence.
        """
        self.dataframe = dataframe
        self.tokenizer = tokenizer
        self.max_input_length = max_input_length
        self.max_target_length = max_target_length

    def __len__(self):
        return len(self.dataframe)

    def __getitem__(self, index):
        # Retrieve the masked function (input) and the target if
condition.
        row = self.dataframe.iloc[index]
        input_text = row['masked_function']
        target_text = row['target_block']

        # Tokenize the input text (i.e., the masked function)
        input_encoding = self.tokenizer(
            input_text,
            truncation=True,
            padding='max_length',
            max_length=self.max_input_length,
            return_tensors="pt"
        )

        # Tokenize the target text (i.e., the original if condition)
        target_encoding = self.tokenizer(
            target_text,
            truncation=True,
            padding='max_length',
            max_length=self.max_target_length,
            return_tensors="pt"
        )

        # Remove the batch dimension added by return_tensors
        input_encoding = {key: val.squeeze(0) for key, val in
input_encoding.items()}
        target_ids = target_encoding['input_ids'].squeeze(0)

        # The labels for training the decoder will be the tokenized
target text.
        input_encoding["labels"] = target_ids

        return input_encoding
```

```
# Example usage for the training dataset:
# (Assuming that df_train is your DataFrame that has been preprocessed
to include the 'masked_function' column)
train_dataset = FineTuneDataset(df_train, tokenizer)
train_dataloader = DataLoader(train_dataset, batch_size=8,
shuffle=True)

# Inspect one batch of tokenized inputs:
for batch in train_dataloader:
    print("Input IDs shape:", batch['input_ids'].shape)
    print("Attention Mask shape:", batch['attention_mask'].shape)
    print("Labels shape:", batch['labels'].shape)
    break  # Just print one batch for verification
```

```
Input IDs shape: torch.Size([8, 512])
Attention Mask shape: torch.Size([8, 512])
Labels shape: torch.Size([8, 128])
```

```
import os
import torch
import gc
from transformers import TrainingArguments, Trainer,
EarlyStoppingCallback, TrainerCallback

# ----------------------#
# GPU Memory Management  #
# ----------------------#
# Use an allocation strategy that reduces fragmentation.
os.environ["PYTORCH_CUDA_ALLOC_CONF"] = "expandable_segments:True"
torch.cuda.empty_cache()
model.gradient_checkpointing_enable()  # Use gradient checkpointing to
lower memory usage during forward passes.

# ----------------------#
# Callbacks              #
# ----------------------#
# Clears GPU cache at the end of each epoch.
class ClearCacheCallback(TrainerCallback):
    def on_epoch_end(self, args, state, control, **kwargs):
        torch.cuda.empty_cache()
        gc.collect()
        return control

# Clears CPU memory after evaluation (freeing any intermediate CPU
tensors).
class ClearEvaluationCallback(TrainerCallback):
    def on_evaluate(self, args, state, control, **kwargs):
        gc.collect()
        return control
```

```
# ----------------------#
# Dataset Initialization #
# ----------------------#
valid_dataset = FineTuneDataset(df_valid, tokenizer)

# ----------------------#
# Training Arguments     #
# ----------------------#
training_args = TrainingArguments(
    output_dir="./codet5_finetuned_gpu",  # Directory for checkpoints
and outputs.
    eval_strategy="epoch",                # Run evaluation at the end
of each epoch.
    save_strategy="epoch",                # Save a checkpoint at the
end of each epoch.
    load_best_model_at_end=True,          # Automatically load the
best model based on eval_loss.
    metric_for_best_model="eval_loss",    # Use evaluation loss for
determining the best model.
    greater_is_better=False,              # Lower evaluation loss is
better.
    num_train_epochs=5,                   # Total number of training
epochs.
    per_device_train_batch_size=64,       # Training batch size per
device.
    per_device_eval_batch_size=64,         # Use the smallest
evaluation batch size to reduce memory use.
    eval_accumulation_steps=256,            # Offload predictions
every 4 steps.
    prediction_loss_only=True,            # Only compute/return loss
to avoid accumulating large prediction tensors.
    learning_rate=5e-5,                   # Learning rate.
    weight_decay=0.01,                    # Weight decay for
regularization.
    fp16=True,                            # Enable FP16 mixed
precision training.
    save_total_limit=2,
    logging_steps=100,
    push_to_hub=False,
)

# ----------------------#
# Metrics Function       #
# ----------------------#
# If only loss is returned, the metric function is not used. If you
need to compute custom metrics,
# consider computing them incrementally to avoid storing full
predictions.
def compute_metrics(eval_pred):
```

```
    predictions, labels = eval_pred
    return {}

# ----------------------#
# Trainer Initialization #
# ----------------------#
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,  # Your training dataset.
    eval_dataset=valid_dataset,   # Your validation dataset.
    processing_class=tokenizer,   # Pass the tokenizer to avoid
deprecation warnings.
    compute_metrics=compute_metrics,
    callbacks=[
        EarlyStoppingCallback(early_stopping_patience=3),
        ClearCacheCallback(),           # Clears GPU cache and
triggers gc.collect() at end of epoch.
        ClearEvaluationCallback()       # Forces CPU garbage
collection after evaluation.
    ]
)

# ----------------------#
# Start Training         #
# ----------------------#
trainer.train()

# Optionally, perform a final garbage collection.
gc.collect()
```

```
wandb: WARNING The `run_name` is currently set to the same value as
`TrainingArguments.output_dir`. If this was not intended, please
specify a different run name by setting the
`TrainingArguments.run_name` parameter.
wandb: Using wandb-core as the SDK backend.  Please refer to
https://wandb.me/wandb-core for more information.

<IPython.core.display.Javascript object>

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: /root/.netrc
wandb: Currently logged in as: bsaurav2004 (bsaurav2004-college-of-
william-mary) to https://api.wandb.ai. Use `wandb login --relogin` to
force relogin

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

`use_cache=True` is incompatible with gradient checkpointing. Setting
`use_cache=False`...

<IPython.core.display.HTML object>

Passing a tuple of `past_key_values` is deprecated and will be removed
in Transformers v4.48.0. You should pass an instance of
`EncoderDecoderCache` instead, e.g.
`past_key_values=EncoderDecoderCache.from_legacy_cache(past_key_values
)`.
There were missing keys in the checkpoint model loaded:
['encoder.embed_tokens.weight', 'decoder.embed_tokens.weight',
'lm_head.weight'].

63

import torch
from torch.utils.data import Dataset, DataLoader
import pandas as pd
import nltk
from nltk.translate.bleu_score import sentence_bleu, SmoothingFunction
from transformers import logging

# NEW: import evaluate for sacrebleu
import evaluate

# 0. Silence Transformers truncation warnings
logging.set_verbosity_error()

# Uncomment if not already downloaded:
# nltk.download('punkt')

################################################
# 1. Define the Dataset Class
################################################
```

```
class FineTuneDataset(Dataset):
    def __init__(self, df, tokenizer, max_length=128,
input_column="cleaned_method"):
        self.df = df
        self.tokenizer = tokenizer
        self.max_length = max_length
        self.input_column = input_column

    def __len__(self):
        return len(self.df)

    def __getitem__(self, idx):
        row = self.df.iloc[idx]
        encoding = self.tokenizer(
            row[self.input_column],
            truncation=True,
            padding="max_length",
            max_length=self.max_length,
            return_tensors="pt",
            verbose=False
        )
        return {k: v.squeeze(0) for k, v in encoding.items()}

print("Step 1: Dataset class defined.")

##############################################
# 2. Load Test Data & Model
##############################################
df_test = pd.read_csv("ft_test.csv")
print("Step 2: Test CSV loaded.")

# Reuse tokenizer & model from your previous cell:
# tokenizer = AutoTokenizer.from_pretrained(model_name)
# model     = AutoModelForSeq2SeqLM.from_pretrained(model_name)

test_dataset = FineTuneDataset(df_test, tokenizer, max_length=128)
test_loader  = DataLoader(test_dataset, batch_size=8, shuffle=False)
model.eval()
print("Step 2: Dataloader ready & model in eval mode.")

##############################################
# 3. Generate Predictions
##############################################
all_predictions = []
with torch.no_grad():
    for batch in test_loader:
        input_ids      = batch["input_ids"].to(model.device)
        attention_mask = batch["attention_mask"].to(model.device)
        generated_ids  = model.generate(
            input_ids=input_ids,
```

```
            attention_mask=attention_mask,
            max_length=128,
            num_beams=5,
            early_stopping=True
        )
        all_predictions += tokenizer.batch_decode(
            generated_ids,
            skip_special_tokens=True,
            clean_up_tokenization_spaces=False
        )

print(f"Step 3: Generated {len(all_predictions)} predictions.")

all_references =
df_test["target_block"].astype(str).str.strip().tolist()
all_inputs     = df_test["cleaned_method"].tolist()
print("Step 3: Extracted references & inputs.")

##############################################
# 4. Align Each Prediction's Token Count to Its Reference
##############################################
pad_tok = tokenizer.pad_token or "<pad>"
aligned_predictions = []
for pred, ref in zip(all_predictions, all_references):
    pred_tokens = pred.split()
    ref_tokens  = ref.split()
    diff = len(ref_tokens) - len(pred_tokens)
    if diff > 0:
        pred_tokens += [pad_tok] * diff
    elif diff < 0:
        pred_tokens = pred_tokens[: len(ref_tokens)]
    aligned_predictions.append(" ".join(pred_tokens))

print("Step 4: Predictions aligned to reference lengths using
pad_token.")

##############################################
# 5. Compute BLEU-4 & Exact Match
##############################################
smooth = SmoothingFunction().method1
bleu_scores = [
    sentence_bleu([ref.split()], pred.split(),
                  weights=(0.25, 0.25, 0.25, 0.25),
                  smoothing_function=smooth) * 100
    for pred, ref in zip(all_predictions, all_references)
]
exact_matches = [pred.strip() == ref for pred, ref in
zip(all_predictions, all_references)]
print("Step 5: BLEU-4 & exact matches computed.")
```

```
##############################################
# 6. Compute SacreBLEU (Corpus-Level)
##############################################
sacrebleu = evaluate.load("sacrebleu")
sacrebleu_results = sacrebleu.compute(
    predictions=all_predictions,
    references=[[r] for r in all_references]
)
corpus_sacrebleu = sacrebleu_results["score"]
print(f"Step 6: Corpus SacreBLEU = {corpus_sacrebleu:.2f}")

##############################################
# 7. Compute Sentence-Level CodeBLEU via SacreBLEU
##############################################
sentence_codebleu_scores = []
for pred, ref in zip(all_predictions, all_references):
    res = sacrebleu.compute(predictions=[pred], references=[[ref]])
    sentence_codebleu_scores.append(res["score"])
print("Step 7: Sentence-level CodeBLEU computed.")

##############################################
# 8. Save Results (with new codebleu_score column)
##############################################
results_df = pd.DataFrame({
    "input":            all_inputs,
    "expected_code":    all_references,
    "predicted_code":   all_predictions,
    "exact_match":      exact_matches,
    "bleu4_score":      bleu_scores,
    "codebleu_score":   sentence_codebleu_scores,
})

results_df.to_csv("testset-results.csv", index=False)
print("Step 8: Results saved to 'testset-results.csv'.")
```

```
Step 1: Dataset class defined.
Step 2: Test CSV loaded.
Step 2: Dataloader ready & model in eval mode.
Step 3: Generated 5000 predictions.
Step 3: Extracted references & inputs.
Step 4: Predictions aligned to reference lengths using pad_token.
Step 5: BLEU-4 & exact matches computed.
```


```
{"model_id":"f8b10552de6247e285bd4cd7d234f8de","version_major":2,"vers
ion_minor":0}
```


```
Step 6: Corpus SacreBLEU = 69.61
Step 7: Sentence-level CodeBLEU computed.
Step 8: Results saved to 'testset-results.csv'.
```

```python
# Cell: Compute, display, and save all evaluation metrics

import pandas as pd
import evaluate

# 1. Load per-example results
df = pd.read_csv("testset-results.csv")
preds = df["predicted_code"].tolist()
refs  = df["expected_code"].tolist()

# 2. Exact Match Rate
exact_rate = df["exact_match"].mean()

# 3. Average BLEU-4 (from the existing column)
avg_bleu4 = df["bleu4_score"].mean()

# 4. SacreBLEU (corpus-level)
sacrebleu = evaluate.load("sacrebleu")
corpus_sacre = sacrebleu.compute(
    predictions=preds,
    references=[[r] for r in refs]
)["score"]

# 4b. Average Sentence-level CodeBLEU (from the new column)
avg_codebleu = df["codebleu_score"].mean()

# 5. Token-level F1
def token_f1(p, r):
    p_tokens = p.split()
    r_tokens = r.split()
    common   = set(p_tokens) & set(r_tokens)
    if not p_tokens or not r_tokens:
        return 0.0
    prec = len(common) / len(p_tokens)
    rec  = len(common) / len(r_tokens)
    return 2 * prec * rec / (prec + rec) if (prec + rec) > 0 else 0.0

df["token_f1"] = df.apply(
    lambda row: token_f1(row["predicted_code"], row["expected_code"]),
    axis=1
)
avg_token_f1 = df["token_f1"].mean()

# 6. Token-level Precision & Recall
def token_precision(p, r):
    p_tokens = p.split()
    if not p_tokens:
        return 0.0
    return len(set(p_tokens) & set(r.split())) / len(p_tokens)
```

```python
def token_recall(p, r):
    r_tokens = r.split()
    if not r_tokens:
        return 0.0
    return len(set(p.split()) & set(r_tokens)) / len(r_tokens)

df["token_precision"] = df.apply(
    lambda row: token_precision(row["predicted_code"],
row["expected_code"]),
    axis=1
)
df["token_recall"] = df.apply(
    lambda row: token_recall(row["predicted_code"],
row["expected_code"]),
    axis=1
)
avg_token_precision = df["token_precision"].mean()
avg_token_recall    = df["token_recall"].mean()

# 7. Average Length Difference
df["length_diff"] = df.apply(
    lambda row: abs(
        len(row["predicted_code"].split()) -
        len(row["expected_code"].split())
    ),
    axis=1
)
avg_length_diff = df["length_diff"].mean()

# 8. Prepare summary lines
summary_lines = [
    f"Exact Match Rate        : {exact_rate:.2%}",
    f"Average BLEU-4          : {avg_bleu4:.2f}",
    f"SacreBLEU (corpus)      : {corpus_sacre:.2f}",
    f"Avg. Sentence CodeBLEU  : {avg_codebleu:.2f}",
    f"Average Token F1        : {avg_token_f1:.2f}",
    f"Average Token Precision : {avg_token_precision:.2f}",
    f"Average Token Recall    : {avg_token_recall:.2f}",
    f"Average Length Diff     : {avg_length_diff:.2f} tokens",
]

# 9. Print to console
for line in summary_lines:
    print(line)

# 10. Write to text file
with open("metrics_summary.txt", "w") as f:
    f.write("\n".join(summary_lines))

print("\nMetrics summary saved to 'metrics_summary.txt'.")
```

```
Exact Match Rate         : 58.98%
Average BLEU-4           : 63.56
SacreBLEU (corpus)       : 69.61
Avg. Sentence CodeBLEU   : 64.07
Average Token F1         : 0.78
Average Token Precision  : 0.80
Average Token Recall     : 0.79
Average Length Diff      : 1.26 tokens

Metrics summary saved to 'metrics_summary.txt'.
```