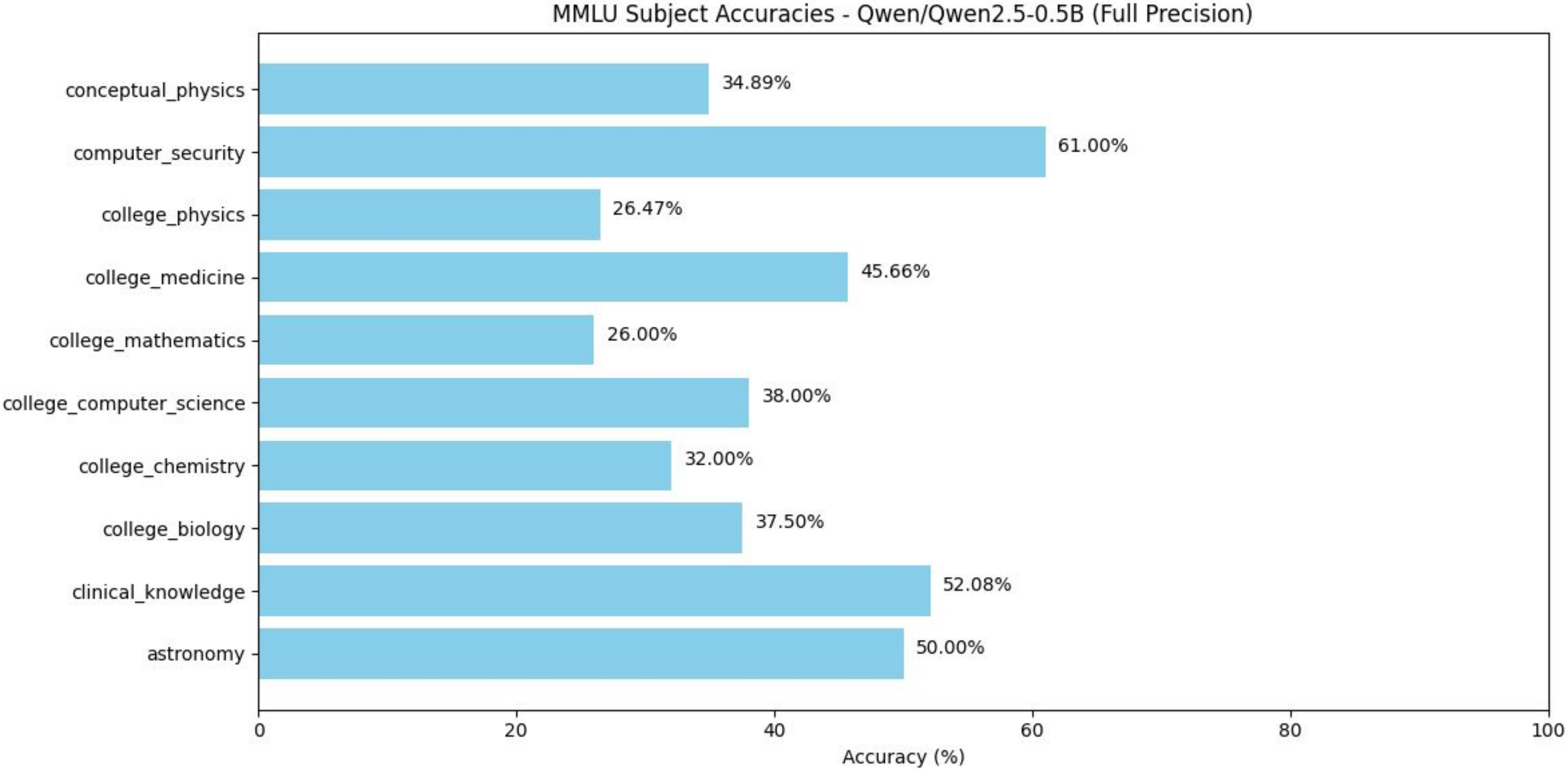
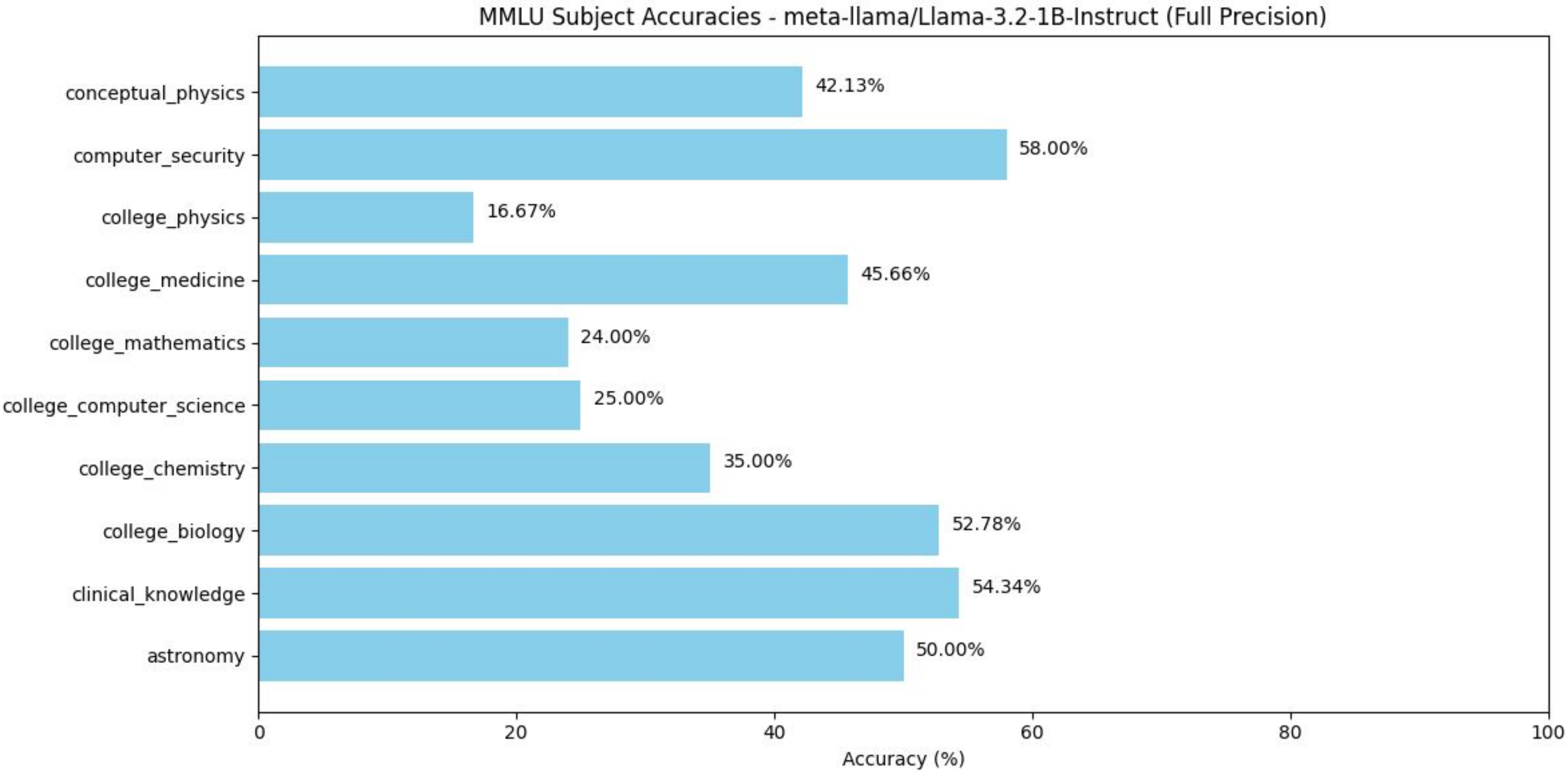

MMLU Subject Accuracies - Qwen/Qwen2.5-0.5B (Full Precision)
conceptual_physics
34.89%
computer_security
61.00%
college_physics
26.47%
college_medicine
45.66%
college_mathematics
26.00%
college_computer_science
38.00%
college_chemistry
32.00%
college_biology
37.50%
clinical_knowledge
52.08%
astronomy
50.00%
0
20
40
60
80
100
Accuracy (%)


MMLU Subject Accuracies - meta-llama/Llama-3.2-1B-Instruct (Full Precision)
conceptual_physics
42.13%
computer_security
58.00%
college_physics
16.67%
college_medicine
45.66%
college_mathematics
24.00%
college_computer_science
25.00%
college_chemistry
35.00%
college_biology
52.78%
clinical_knowledge
54.34%
astronomy
50.00%
0
20
40
60
80
100
Accuracy (%)

```
# 1. Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# 2. Create and move into a specific project folder
import os
project_folder = '/content/drive/MyDrive/Colab_Projects/RunningLMM'

if not os.path.exists(project_folder):
    os.makedirs(project_folder)
    print(f"Created folder: {project_folder}")

# 3. Change the working directory to this folder
os.chdir(project_folder)
print(f"Current Directory: {os.getcwd()}")
```

```
Mounted at /content/drive
Current Directory: /content/drive/MyDrive/Colab_Projects/RunningLMM
```

```
# Run this in a cell
!nvidia-smi
```

```
Wed Jan 14 19:37:17 2026
+---------------------------------------------------------------------
--------------------+
| NVIDIA-SMI 550.54.15               Driver Version: 550.54.15
CUDA Version: 12.4     |
|-----------------------------------------+------------------------
+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A |
Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage |
GPU-Util  Compute M. |
|                                         |                        |
MIG M. |
|
=========================================+========================+===
===================|
|   0  Tesla T4                       Off |   00000000:00:04.0 Off |
0 |
| N/A   47C    P8             10W /   70W |       0MiB /  15360MiB |
0%      Default |
|                                         |                        |
N/A |
+-----------------------------------------+------------------------
+----------------------+


+---------------------------------------------------------------------
--------------------+
| Processes:
|
|  GPU   GI   CI        PID   Type   Process name
GPU Memory |
|        ID   ID
Usage      |
|
======================================================================
===================|
|  No running processes found
|
+---------------------------------------------------------------------
--------------------+
```

```
!curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash - &&
\
sudo apt-get install -y nodejs && \
sudo npm install -g @anthropic-ai/claude-code && \
export PATH=/usr/bin:$PATH
```

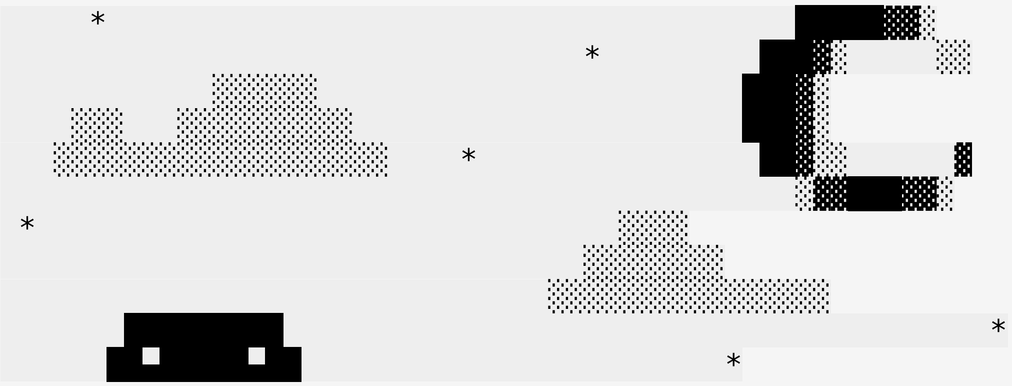
```
2026-01-14 19:37:42 - Installing pre-requisites
Get:1
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/
x86_64  InRelease [1,581 B]
Get:2 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/
InRelease [3,632 B]
Get:3 https://cli.github.com/packages stable InRelease [3,917 B]
Get:4
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/
x86_64  Packages [2,297 kB]
Hit:5 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:6 http://security.ubuntu.com/ubuntu jammy-security InRelease [129
kB]
Get:7 https://r2u.stat.illinois.edu/ubuntu jammy InRelease [6,555 B]
Get:8 https://cli.github.com/packages stable/main amd64 Packages [354
B]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128
kB]
Get:10 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages
[9,606 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/restricted
amd64 Packages [6,205 kB]
Hit:12 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy
InRelease
Get:13 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127
kB]
Hit:14 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu
jammy InRelease
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64
Packages [3,968 kB]
```

```
Hit:16 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy
InRelease
Get:17 https://r2u.stat.illinois.edu/ubuntu jammy/main amd64 Packages
[2,868 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64
Packages [1,600 kB]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/main amd64
Packages [3,637 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64
Packages [69.2 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64
Packages [6,411 kB]
Get:22 http://security.ubuntu.com/ubuntu jammy-security/universe amd64
Packages [1,289 kB]
Get:23 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64
Packages [37.2 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64
Packages [83.9 kB]
Fetched 38.5 MB in 4s (8,874 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
90 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: Skipping acquire of configured file 'main/source/Sources' as
repository 'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does
not seem to provide it (sources.list entry misspelt?)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203~22.04.1).
curl is already the newest version (7.81.0-1ubuntu1.21).
The following additional packages will be installed:
  dirmngr gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-
server
  gpgconf gpgsm gpgv
Suggested packages:
  pinentry-gnome3 tor parcimonie xloadimage scdaemon
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client
  gpg-wks-server gpgconf gpgsm gpgv
11 upgraded, 1 newly installed, 0 to remove and 79 not upgraded.
Need to get 2,249 kB of archives.
After this operation, 170 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 gpg-
wks-client amd64 2.2.27-3ubuntu2.5 [62.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64
dirmngr amd64 2.2.27-3ubuntu2.5 [293 kB]
```

```
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 gpg-
wks-server amd64 2.2.27-3ubuntu2.5 [57.6 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 gnupg-
utils amd64 2.2.27-3ubuntu2.5 [309 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 gpg-
agent amd64 2.2.27-3ubuntu2.5 [209 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 gpg
amd64 2.2.27-3ubuntu2.5 [519 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64
gpgconf amd64 2.2.27-3ubuntu2.5 [94.3 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 gnupg-
l10n all 2.2.27-3ubuntu2.5 [54.5 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 gnupg
all 2.2.27-3ubuntu2.5 [315 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 gpgsm
amd64 2.2.27-3ubuntu2.5 [197 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 gpgv
amd64 2.2.27-3ubuntu2.5 [137 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64
apt-transport-https all 2.4.14 [1,510 B]
Fetched 2,249 kB in 1s (1,560 kB/s)
(Reading database ... 121689 files and directories currently
installed.)
Preparing to unpack .../00-gpg-wks-client_2.2.27-
3ubuntu2.5_amd64.deb ...
Unpacking gpg-wks-client (2.2.27-3ubuntu2.5) over (2.2.27-
3ubuntu2.4) ...
Preparing to unpack .../01-dirmngr_2.2.27-3ubuntu2.5_amd64.deb ...
Unpacking dirmngr (2.2.27-3ubuntu2.5) over (2.2.27-3ubuntu2.4) ...
Preparing to unpack .../02-gpg-wks-server_2.2.27-
3ubuntu2.5_amd64.deb ...
Unpacking gpg-wks-server (2.2.27-3ubuntu2.5) over (2.2.27-
3ubuntu2.4) ...
Preparing to unpack .../03-gnupg-utils_2.2.27-3ubuntu2.5_amd64.deb ...
Unpacking gnupg-utils (2.2.27-3ubuntu2.5) over (2.2.27-3ubuntu2.4) ...
Preparing to unpack .../04-gpg-agent_2.2.27-3ubuntu2.5_amd64.deb ...
Unpacking gpg-agent (2.2.27-3ubuntu2.5) over (2.2.27-3ubuntu2.4) ...
Preparing to unpack .../05-gpg_2.2.27-3ubuntu2.5_amd64.deb ...
Unpacking gpg (2.2.27-3ubuntu2.5) over (2.2.27-3ubuntu2.4) ...
Preparing to unpack .../06-gpgconf_2.2.27-3ubuntu2.5_amd64.deb ...
Unpacking gpgconf (2.2.27-3ubuntu2.5) over (2.2.27-3ubuntu2.4) ...
Preparing to unpack .../07-gnupg-l10n_2.2.27-3ubuntu2.5_all.deb ...
Unpacking gnupg-l10n (2.2.27-3ubuntu2.5) over (2.2.27-3ubuntu2.4) ...
Preparing to unpack .../08-gnupg_2.2.27-3ubuntu2.5_all.deb ...
Unpacking gnupg (2.2.27-3ubuntu2.5) over (2.2.27-3ubuntu2.4) ...
Preparing to unpack .../09-gpgsm_2.2.27-3ubuntu2.5_amd64.deb ...
Unpacking gpgsm (2.2.27-3ubuntu2.5) over (2.2.27-3ubuntu2.4) ...
Preparing to unpack .../10-gpgv_2.2.27-3ubuntu2.5_amd64.deb ...
Unpacking gpgv (2.2.27-3ubuntu2.5) over (2.2.27-3ubuntu2.4) ...
```

```
Setting up gpgv (2.2.27-3ubuntu2.5) ...
Selecting previously unselected package apt-transport-https.
(Reading database ... 121689 files and directories currently
installed.)
Preparing to unpack .../apt-transport-https_2.4.14_all.deb ...
Unpacking apt-transport-https (2.4.14) ...
Setting up apt-transport-https (2.4.14) ...
Setting up gnupg-l10n (2.2.27-3ubuntu2.5) ...
Setting up gpgconf (2.2.27-3ubuntu2.5) ...
Setting up gpg (2.2.27-3ubuntu2.5) ...
Setting up gnupg-utils (2.2.27-3ubuntu2.5) ...
Setting up gpg-agent (2.2.27-3ubuntu2.5) ...
Setting up gpgsm (2.2.27-3ubuntu2.5) ...
Setting up dirmngr (2.2.27-3ubuntu2.5) ...
Setting up gpg-wks-server (2.2.27-3ubuntu2.5) ...
Setting up gpg-wks-client (2.2.27-3ubuntu2.5) ...
Setting up gnupg (2.2.27-3ubuntu2.5) ...
Processing triggers for man-db (2.10.2-1) ...
Hit:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/
InRelease
Hit:2 https://cli.github.com/packages stable InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/
x86_64  InRelease
Get:5 https://deb.nodesource.com/node_20.x nodistro InRelease [12.1
kB]
Hit:6 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:7 https://r2u.stat.illinois.edu/ubuntu jammy InRelease
Hit:8 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:9 https://deb.nodesource.com/node_20.x nodistro/main amd64
Packages [13.6 kB]
Hit:10 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:11 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy
InRelease
Hit:12 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu
jammy InRelease
Hit:13 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy
InRelease
Fetched 25.8 kB in 1s (18.9 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
79 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: Skipping acquire of configured file 'main/source/Sources' as
repository 'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does
not seem to provide it (sources.list entry misspelt?)
2026-01-14 19:38:02 - Repository configured successfully.
2026-01-14 19:38:02 - To install Node.js, run: apt install nodejs -y
```

```
2026-01-14 19:38:02 - You can use N|solid Runtime as a node.js alternative
2026-01-14 19:38:02 - To install N|solid Runtime, run: apt install nsolid -y

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 79 not upgraded.
Need to get 32.0 MB of archives.
After this operation, 197 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_20.x nodistro/main amd64 nodejs amd64 20.20.0-1nodesource1 [32.0 MB]
Fetched 32.0 MB in 0s (72.8 MB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78, <> line 1.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package nodejs.
(Reading database ... 121693 files and directories currently installed.)
Preparing to unpack .../nodejs_20.20.0-1nodesource1_amd64.deb ...
Unpacking nodejs (20.20.0-1nodesource1) ...
Setting up nodejs (20.20.0-1nodesource1) ...
Processing triggers for man-db (2.10.2-1) ...
 fund` for details

!claude

Welcome to Claude Code v2.1.7
```



```
                     *
…………………………………………………………………………………………………………………………………………

Choose the text style that looks best with your terminal
 To change this later, run /theme

 ❯ 1. Dark mode ✔
   2. Light mode
   3. Dark mode (colorblind-friendly)
   4. Light mode (colorblind-friendly)
   5. Dark mode (ANSI colors only)
   6. Light mode (ANSI colors only)

----------------------------------------------------------------------
----------
 1   function greet() {

 2 -  console.log("Hello, World!");

 2 +  console.log("Hello, Claude!");

 3   }

----------------------------------------------------------------------
----------
 Syntax highlighting available only in native build
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 59.1/59.1 MB 17.1 MB/s eta
0:00:00

!pip install -q transformers torch datasets accelerate tqdm
huggingface_hub bitsandbytes

!hf auth login


    _|    _|  _|    _|    _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|
_|      _|_|_|_|    _|_|      _|_|_|  _|_|_|_|
    _|    _|  _|    _|  _|        _|          _|    _|_|    _|  _|
_|        _|    _|  _|        _|
    _|_|_|_|  _|    _|  _|  _|_|  _|  _|_|    _|    _|  _|  _|  _|  _|
_|      _|_|_|    _|_|_|_|  _|        _|_|_|
    _|    _|  _|    _|  _|    _|  _|    _|    _|    _|    _|_|  _|
_|      _|        _|    _|  _|        _|
    _|    _|    _|_|      _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|
_|      _|        _|    _|    _|_|_|  _|_|_|_|

    To log in, `huggingface_hub` requires a token generated from
https://huggingface.co/settings/tokens .
Enter your token (input will not be visible):
Add token as git credential? (Y/n) y
Token is valid (permission: fineGrained).
```

```
The token `agentic_token` has been saved to
/root/.cache/huggingface/stored_tokens
Cannot authenticate through git-credential as no helper is defined on
your machine.
You might have to re-authenticate when pushing to the Hugging Face
Hub.
Run the following command in your terminal in case you want to set the
'store' credential helper as default.

git config --global credential.helper store

Read https://git-scm.com/book/en/v2/Git-Tools-Credential-Storage for
more details.
Token has not been saved to git credential helper.
Your token has been saved to /root/.cache/huggingface/token
Login successful.
The current active token is: `agentic_token`
```

Various Quantized Models

GPU & No Quantization

Real Time: 0.4 minutes

CPU Time (process): 0.2 minutes

GPU Time (CUDA kernels): 0.3 minutes

Inferences timed: 252

```
!python llama_mmlu_eval_quantized.py


======================================================================
Llama 3.2-1B MMLU Evaluation (Quantized)
======================================================================

======================================================================
Environment Check
======================================================================
 Running in Google Colab
 Platform: Linux (x86_64)
 GPU Available: Tesla T4
 GPU Memory: 15.83 GB
 Quantization disabled - loading full precision model
 Hugging Face authenticated
```

```
======================================================================
Configuration
======================================================================
Model: meta-llama/Llama-3.2-1B-Instruct
Device: cuda
Quantization: None (full precision)
Expected memory: ~2.5 GB (FP16)
Number of subjects: 2
======================================================================


Loading model meta-llama/Llama-3.2-1B-Instruct...
Device: cuda
 Tokenizer loaded
Loading model (this may take 2-3 minutes)...
2026-01-14 20:08:25.901823: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to
register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
WARNING: All log messages before absl::InitializeLog() is called are
written to STDERR
E0000 00:00:1768421305.978845   11502 cuda_dnn.cc:8579] Unable to
register cuDNN factory: Attempting to register factory for plugin
cuDNN when one has already been registered
E0000 00:00:1768421305.996178   11502 cuda_blas.cc:1407] Unable to
register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
W0000 00:00:1768421306.055758   11502 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768421306.055791   11502 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768421306.055799   11502 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768421306.055805   11502 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
2026-01-14 20:08:26.068130: I
tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow
binary is optimized to use available CPU instructions in performance-
critical operations.
To enable the following instructions: AVX2 FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
 Model loaded successfully!
  Model device: cuda:0
  Model dtype: torch.float16
  GPU Memory: 2.47 GB allocated, 2.51 GB reserved
```

```
======================================================================
Starting evaluation on 2 subjects
======================================================================


Progress: 1/2 subjects

======================================================================
Evaluating subject: astronomy
======================================================================
Testing astronomy:   0% 0/152 [00:00<?, ?it/s]The following generation
flags are not valid and may be ignored: ['top_p']. Set
`TRANSFORMERS_VERBOSITY=info` for more details.
Testing astronomy: 100% 152/152 [00:09<00:00, 15.63it/s]
 Result: 75/152 correct = 49.34%

Progress: 2/2 subjects

======================================================================
Evaluating subject: business_ethics
======================================================================
business_ethics/test-00000-of-00001.parq(…): 100% 21.6k/21.6k
[00:00<00:00, 23.3kB/s]
business_ethics/validation-00000-of-0000(…): 100% 5.09k/5.09k
[00:00<00:00, 15.2kB/s]
business_ethics/dev-00000-of-00001.parqu(…): 100% 4.96k/4.96k
[00:00<00:00, 10.6kB/s]
Generating test split: 100% 100/100 [00:00<00:00, 1196.31 examples/s]
Generating validation split: 100% 11/11 [00:00<00:00, 2188.06
examples/s]
Generating dev split: 100% 5/5 [00:00<00:00, 1810.07 examples/s]
Testing business_ethics: 100% 100/100 [00:06<00:00, 16.08it/s]
 Result: 45/100 correct = 45.00%

======================================================================
EVALUATION SUMMARY
======================================================================
Model: meta-llama/Llama-3.2-1B-Instruct
None (full precision)
Total Subjects: 2
Total Questions: 252
Total Correct: 120
Overall Accuracy: 47.62%
Real Time: 0.4 minutes
CPU Time (process): 0.2 minutes
GPU Time (CUDA kernels): 0.3 minutes
Inferences timed: 252
======================================================================

 Results saved to: llama_3.2_1b_mmlu_results_full_20260114_200906.json
```

```

ðŸ“Š Top 5 Subjects:
  1. astronomy: 49.34%
  2. business_ethics: 45.00%

ðŸ“‰ Bottom 5 Subjects:
  1. astronomy: 49.34%
  2. business_ethics: 45.00%

======================================================================
ðŸ’¾ To download results in Colab:
======================================================================
from google.colab import files
files.download('llama_3.2_1b_mmlu_results_full_20260114_200906.json')
Figure(1200x600)

ðŸ“· Accuracy plot saved to:
llama_3.2_1b_mmlu_accuracies_full_20260114_200906.png

âœ… Evaluation complete!
```

GPU & 4 Bit

Real Time: 0.3 minutes

CPU Time (process): 0.3 minutes

GPU Time (CUDA kernels): 0.3 minutes

Inferences timed: 252

```
!python llama_mmlu_eval_quantized.py


======================================================================
Llama 3.2-1B MMLU Evaluation (Quantized)
======================================================================

======================================================================
Environment Check
======================================================================
 Running in Google Colab
 Platform: Linux (x86_64)
 GPU Available: Tesla T4
 GPU Memory: 15.83 GB
 bitsandbytes installed - 4-bit quantization available
 Hugging Face authenticated

======================================================================
Configuration
======================================================================
Model: meta-llama/Llama-3.2-1B-Instruct
Device: cuda
Quantization: 4-bit
Expected memory: ~1.5 GB
Number of subjects: 2
======================================================================


Loading model meta-llama/Llama-3.2-1B-Instruct...
Device: cuda
 Tokenizer loaded
Using 4-bit quantization (NF4 + double quant)
Memory usage: ~1.5 GB
Loading model (this may take 2-3 minutes)...
2026-01-14 20:09:47.950502: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to
register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
WARNING: All log messages before absl::InitializeLog() is called are
written to STDERR
E0000 00:00:1768421387.970815   11913 cuda_dnn.cc:8579] Unable to
register cuDNN factory: Attempting to register factory for plugin
cuDNN when one has already been registered
E0000 00:00:1768421387.975742   11913 cuda_blas.cc:1407] Unable to
register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
W0000 00:00:1768421387.988446   11913 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768421387.988468   11913 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768421387.988472   11913 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768421387.988478   11913 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
2026-01-14 20:09:47.992454: I
tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow
binary is optimized to use available CPU instructions in performance-
critical operations.
To enable the following instructions: AVX2 FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
 Model loaded successfully!
  Model device: cuda:0
  Model dtype: torch.float16
  GPU Memory: 1.03 GB allocated, 1.27 GB reserved
  Quantization: 4-bit active
```

```
======================================================================
Starting evaluation on 2 subjects
======================================================================


Progress: 1/2 subjects

======================================================================
Evaluating subject: astronomy
======================================================================
Testing astronomy:   0% 0/152 [00:00<?, ?it/s]The following generation
flags are not valid and may be ignored: ['top_p']. Set
`TRANSFORMERS_VERBOSITY=info` for more details.
Testing astronomy: 100% 152/152 [00:10<00:00, 14.39it/s]
 Result: 72/152 correct = 47.37%

Progress: 2/2 subjects

======================================================================
Evaluating subject: business_ethics
======================================================================
Testing business_ethics: 100% 100/100 [00:07<00:00, 12.63it/s]
 Result: 40/100 correct = 40.00%

======================================================================
EVALUATION SUMMARY
======================================================================
Model: meta-llama/Llama-3.2-1B-Instruct
Quantization: 4-bit
Total Subjects: 2
Total Questions: 252
Total Correct: 112
Overall Accuracy: 44.44%
Real Time: 0.3 minutes
CPU Time (process): 0.3 minutes
GPU Time (CUDA kernels): 0.3 minutes
Inferences timed: 252
======================================================================

 Results saved to: llama_3.2_1b_mmlu_results_4bit_20260114_201018.json

ðŸ“Š Top 5 Subjects:
  1. astronomy: 47.37%
  2. business_ethics: 40.00%

ðŸ“‰ Bottom 5 Subjects:
  1. astronomy: 47.37%
  2. business_ethics: 40.00%

======================================================================
ðŸ’¾ To download results in Colab:
======================================================================
from google.colab import files
files.download('llama_3.2_1b_mmlu_results_4bit_20260114_201018.json')
Figure(1200x600)

ðŸ“· Accuracy plot saved to:
llama_3.2_1b_mmlu_accuracies_4bit_20260114_201018.png

âœ… Evaluation complete!
```

GPU & 8 Bit

Real Time: 0.5 minutes CPU Time (process): 0.5 minutes GPU Time (CUDA kernels): 0.5 minutes

```
!python llama_mmlu_eval_quantized.py


======================================================================
Llama 3.2-1B MMLU Evaluation (Quantized)
======================================================================

======================================================================
Environment Check
======================================================================
 Running in Google Colab
 Platform: Linux (x86_64)
 GPU Available: Tesla T4
 GPU Memory: 15.83 GB
 bitsandbytes installed - 8-bit quantization available
 Hugging Face authenticated

======================================================================
Configuration
======================================================================
Model: meta-llama/Llama-3.2-1B-Instruct
Device: cuda
Quantization: 8-bit
Expected memory: ~2.5 GB
Number of subjects: 2
======================================================================


Loading model meta-llama/Llama-3.2-1B-Instruct...
Device: cuda
 Tokenizer loaded
Using 8-bit quantization
Memory usage: ~2.5 GB
Loading model (this may take 2-3 minutes)...
2026-01-14 20:11:10.152507: E
```

```
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to
register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
WARNING: All log messages before absl::InitializeLog() is called are
written to STDERR
E0000 00:00:1768421470.175182   12276 cuda_dnn.cc:8579] Unable to
register cuDNN factory: Attempting to register factory for plugin
cuDNN when one has already been registered
E0000 00:00:1768421470.180197   12276 cuda_blas.cc:1407] Unable to
register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
W0000 00:00:1768421470.193422   12276 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768421470.193448   12276 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768421470.193452   12276 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768421470.193457   12276 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
2026-01-14 20:11:10.197360: I
tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow
binary is optimized to use available CPU instructions in performance-
critical operations.
To enable the following instructions: AVX2 FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
 Model loaded successfully!
  Model device: cuda:0
  Model dtype: torch.float16
  GPU Memory: 1.50 GB allocated, 1.59 GB reserved
  Quantization: 8-bit active

======================================================================
Starting evaluation on 2 subjects
======================================================================


Progress: 1/2 subjects

======================================================================
Evaluating subject: astronomy
======================================================================
Testing astronomy:   0% 0/152 [00:00<?, ?it/s]The following generation
flags are not valid and may be ignored: ['top_p']. Set
`TRANSFORMERS_VERBOSITY=info` for more details.
Testing astronomy: 100% 152/152 [00:18<00:00,  8.40it/s]
 Result: 76/152 correct = 50.00%
```

```
Progress: 2/2 subjects

======================================================================
Evaluating subject: business_ethics
======================================================================
Testing business_ethics: 100% 100/100 [00:11<00:00,  8.76it/s]
 Result: 45/100 correct = 45.00%

======================================================================
EVALUATION SUMMARY
======================================================================
Model: meta-llama/Llama-3.2-1B-Instruct
Quantization: 8-bit
Total Subjects: 2
Total Questions: 252
Total Correct: 121
Overall Accuracy: 48.02%
Real Time: 0.5 minutes
CPU Time (process): 0.5 minutes
GPU Time (CUDA kernels): 0.5 minutes
Inferences timed: 252
======================================================================

 Results saved to: llama_3.2_1b_mmlu_results_8bit_20260114_201150.json

ðŸ“Š Top 5 Subjects:
  1. astronomy: 50.00%
  2. business_ethics: 45.00%

ðŸ“‰ Bottom 5 Subjects:
  1. astronomy: 50.00%
  2. business_ethics: 45.00%

======================================================================
ðŸ’¾ To download results in Colab:
======================================================================
from google.colab import files
files.download('llama_3.2_1b_mmlu_results_8bit_20260114_201150.json')
Figure(1200x600)

ðŸ“· Accuracy plot saved to:
llama_3.2_1b_mmlu_accuracies_8bit_20260114_201150.png

âœ… Evaluation complete!
```

CPU Models ran too slow

Two Additional Models

```
!python llama_mmlu_eval_quantized.py


======================================================================
Llama 3.2-1B MMLU Evaluation (Quantized)
======================================================================

======================================================================
Environment Check
======================================================================
 Running in Google Colab
 Platform: Linux (x86_64)
 GPU Available: Tesla T4
 GPU Memory: 15.83 GB
 Quantization disabled - loading full precision model
 Hugging Face authenticated

======================================================================
Configuration
======================================================================
Model: allenai/OLMo-2-0425-1B
Device: cuda
Quantization: None (full precision)
Expected memory: ~2.5 GB (FP16)
Number of subjects: 10
======================================================================


Loading model allenai/OLMo-2-0425-1B...
Device: cuda
tokenizer_config.json: 4.34kB [00:00, 12.5MB/s]
vocab.json: 1.61MB [00:00, 25.6MB/s]
merges.txt: 917kB [00:00, 14.2MB/s]
tokenizer.json: 7.14MB [00:00, 28.5MB/s]
special_tokens_map.json: 100% 125/125 [00:00<00:00, 620kB/s]
 Tokenizer loaded
Loading model (this may take 2-3 minutes)...
config.json: 100% 623/623 [00:00<00:00, 4.41MB/s]
2026-01-14 19:52:23.554892: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to
register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
WARNING: All log messages before absl::InitializeLog() is called are
written to STDERR
E0000 00:00:1768420343.590921    7064 cuda_dnn.cc:8579] Unable to
```

```
register cuDNN factory: Attempting to register factory for plugin
cuDNN when one has already been registered
E0000 00:00:1768420343.600786    7064 cuda_blas.cc:1407] Unable to
register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
W0000 00:00:1768420343.623378    7064 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768420343.623431    7064 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768420343.623440    7064 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768420343.623447    7064 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
2026-01-14 19:52:23.630460: I
tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow
binary is optimized to use available CPU instructions in performance-
critical operations.
To enable the following instructions: AVX2 FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
model.safetensors.index.json: 14.9kB [00:00, 48.2MB/s]
Fetching 2 files:   0% 0/2 [00:00<?, ?it/s]
model-00002-of-00002.safetensors:   0% 0.00/956M [00:00<?, ?B/s]odel-
00001-of-00002.safetensors:   0% 0.00/4.98G [00:00<?, ?B/s]odel-00001-
of-00002.safetensors:   0% 628k/4.98G [00:02<5:15:02, 264kB/s]odel-
00001-of-00002.safetensors:   0% 1.94M/4.98G [00:03<2:32:30,
544kB/s]odel-00002-of-00002.safetensors:   7% 67.1M/956M [00:04<01:04,
13.7MB/s]odel-00002-of-00002.safetensors:   9% 84.8M/956M
[00:05<00:49, 17.6MB/s]odel-00002-of-00002.safetensors:  16% 152M/956M
[00:05<00:23, 33.6MB/s] odel-00001-of-00002.safetensors:   1%
69.0M/4.98G [00:06<05:45, 14.2MB/s] odel-00001-of-00002.safetensors:
3% 136M/4.98G [00:06<02:33, 31.6MB/s] odel-00001-of-00002.safetensors:
4% 203M/4.98G [00:07<01:33, 51.0MB/s]odel-00001-of-00002.safetensors:
5% 270M/4.98G [00:07<01:09, 67.8MB/s]odel-00001-of-00002.safetensors:
7% 337M/4.98G [00:07<00:52, 88.8MB/s]odel-00002-of-00002.safetensors:
23% 219M/956M [00:10<00:36, 20.4MB/s]odel-00002-of-00002.safetensors:
30% 286M/956M [00:11<00:23, 29.1MB/s]odel-00002-of-00002.safetensors:
37% 353M/956M [00:11<00:13, 43.1MB/s]odel-00002-of-00002.safetensors:
44% 420M/956M [00:12<00:11, 47.0MB/s]odel-00001-of-00002.safetensors:
8% 404M/4.98G [00:13<02:38, 28.9MB/s]odel-00002-of-00002.safetensors:
51% 487M/956M [00:16<00:15, 31.1MB/s]odel-00002-of-00002.safetensors:
58% 554M/956M [00:16<00:09, 42.3MB/s]odel-00002-of-00002.safetensors:
65% 621M/956M [00:17<00:06, 54.7MB/s]odel-00002-of-00002.safetensors:
72% 688M/956M [00:17<00:04, 66.0MB/s]odel-00001-of-00002.safetensors:
10% 473M/4.98G [00:18<03:29, 21.6MB/s]odel-00002-of-00002.safetensors:
79% 755M/956M [00:18<00:02, 68.1MB/s]odel-00001-of-00002.safetensors:
```

```
11% 541M/4.98G [00:19<02:47, 26.6MB/s]odel-00002-of-00002.safetensors:
86% 822M/956M [00:19<00:01, 76.4MB/s]odel-00002-of-00002.safetensors:
93% 889M/956M [00:22<00:01, 42.4MB/s]odel-00001-of-00002.safetensors:
12% 608M/4.98G [00:22<03:02, 24.0MB/s]odel-00002-of-00002.safetensors:
100% 956M/956M [00:23<00:00, 40.2MB/s]


model-00001-of-00002.safetensors:  14% 675M/4.98G [00:24<02:31,
28.5MB/s]odel-00001-of-00002.safetensors:  15% 742M/4.98G
[00:24<01:48, 39.2MB/s]odel-00001-of-00002.safetensors:  16%
809M/4.98G [00:26<01:51, 37.4MB/s]odel-00001-of-00002.safetensors:
18% 876M/4.98G [00:26<01:24, 48.5MB/s]odel-00001-of-00002.safetensors:
19% 943M/4.98G [00:27<01:03, 63.9MB/s]odel-00001-of-00002.safetensors:
20% 1.01G/4.98G [00:27<00:49, 80.6MB/s]odel-00001-of-
00002.safetensors:  22% 1.08G/4.98G [00:28<00:45, 85.9MB/s]odel-00001-
of-00002.safetensors:  23% 1.14G/4.98G [00:28<00:37, 103MB/s] odel-
00001-of-00002.safetensors:  24% 1.21G/4.98G [00:29<00:44,
84.9MB/s]odel-00001-of-00002.safetensors:  26% 1.28G/4.98G
[00:30<00:41, 90.0MB/s]odel-00001-of-00002.safetensors:  27%
1.35G/4.98G [00:33<01:14, 48.5MB/s]odel-00001-of-00002.safetensors:
28% 1.41G/4.98G [00:33<00:55, 64.4MB/s]odel-00001-of-
00002.safetensors:  30% 1.48G/4.98G [00:33<00:44, 79.3MB/s]odel-00001-
of-00002.safetensors:  31% 1.55G/4.98G [00:39<01:50, 31.0MB/s]odel-
00001-of-00002.safetensors:  32% 1.62G/4.98G [00:40<01:44,
32.2MB/s]odel-00001-of-00002.safetensors:  34% 1.68G/4.98G
[00:41<01:16, 42.9MB/s]odel-00001-of-00002.safetensors:  35%
1.75G/4.98G [00:41<00:58, 55.4MB/s]odel-00001-of-00002.safetensors:
36% 1.82G/4.98G [00:42<00:55, 57.4MB/s]odel-00001-of-
00002.safetensors:  38% 1.89G/4.98G [00:43<00:42, 73.7MB/s]odel-00001-
of-00002.safetensors:  39% 1.95G/4.98G [00:43<00:31, 95.9MB/s]odel-
00001-of-00002.safetensors:  41% 2.02G/4.98G [00:43<00:25, 115MB/s]
odel-00001-of-00002.safetensors:  42% 2.09G/4.98G [00:44<00:30,
94.5MB/s]odel-00001-of-00002.safetensors:  43% 2.15G/4.98G
[00:49<01:21, 34.9MB/s]odel-00001-of-00002.safetensors:  45%
2.24G/4.98G [00:50<01:10, 39.2MB/s]odel-00001-of-00002.safetensors:
46% 2.31G/4.98G [00:51<00:58, 46.1MB/s]odel-00001-of-
00002.safetensors:  48% 2.37G/4.98G [00:53<00:54, 48.0MB/s]odel-00001-
of-00002.safetensors:  49% 2.44G/4.98G [00:53<00:44, 56.9MB/s]odel-
00001-of-00002.safetensors:  50% 2.51G/4.98G [00:54<00:34,
71.0MB/s]odel-00001-of-00002.safetensors:  52% 2.57G/4.98G
[00:54<00:28, 83.4MB/s]odel-00001-of-00002.safetensors:  53%
2.64G/4.98G [00:54<00:24, 97.0MB/s]odel-00001-of-00002.safetensors:
54% 2.71G/4.98G [00:55<00:18, 126MB/s] odel-00001-of-
00002.safetensors:  56% 2.77G/4.98G [00:56<00:27, 79.6MB/s]odel-00001-
of-00002.safetensors:  57% 2.84G/4.98G [00:57<00:26, 80.9MB/s]odel-
00001-of-00002.safetensors:  58% 2.91G/4.98G [00:59<00:36,
56.2MB/s]odel-00001-of-00002.safetensors:  60% 2.97G/4.98G
[00:59<00:26, 76.2MB/s]odel-00001-of-00002.safetensors:  61%
3.04G/4.98G [01:00<00:23, 83.7MB/s]odel-00001-of-00002.safetensors:
```

```
62% 3.11G/4.98G [01:00<00:18, 101MB/s] odel-00001-of-
00002.safetensors:  64% 3.18G/4.98G [01:05<00:53, 34.0MB/s]odel-00001-
of-00002.safetensors:  65% 3.24G/4.98G [01:05<00:37, 46.0MB/s]odel-
00001-of-00002.safetensors:  66% 3.31G/4.98G [01:06<00:26,
62.7MB/s]odel-00001-of-00002.safetensors:  68% 3.38G/4.98G
[01:06<00:21, 75.0MB/s]odel-00001-of-00002.safetensors:  69%
3.44G/4.98G [01:09<00:33, 46.4MB/s]odel-00001-of-00002.safetensors:
70% 3.51G/4.98G [01:09<00:24, 59.5MB/s]odel-00001-of-
00002.safetensors:  72% 3.58G/4.98G [01:09<00:18, 77.6MB/s]odel-00001-
of-00002.safetensors:  73% 3.64G/4.98G [01:10<00:14, 93.1MB/s]odel-
00001-of-00002.safetensors:  74% 3.71G/4.98G [01:10<00:10, 119MB/s]
odel-00001-of-00002.safetensors:  76% 3.78G/4.98G [01:11<00:11,
109MB/s]odel-00001-of-00002.safetensors:  77% 3.85G/4.98G
[01:11<00:08, 132MB/s]odel-00001-of-00002.safetensors:  79%
3.91G/4.98G [01:11<00:07, 144MB/s]odel-00001-of-00002.safetensors:
80% 3.98G/4.98G [01:12<00:06, 149MB/s]odel-00001-of-00002.safetensors:
81% 4.05G/4.98G [01:12<00:06, 150MB/s]odel-00001-of-00002.safetensors:
83% 4.11G/4.98G [01:13<00:07, 124MB/s]odel-00001-of-00002.safetensors:
84% 4.18G/4.98G [01:13<00:05, 154MB/s]odel-00001-of-00002.safetensors:
85% 4.25G/4.98G [01:14<00:05, 134MB/s]odel-00001-of-00002.safetensors:
87% 4.31G/4.98G [01:14<00:04, 148MB/s]odel-00001-of-00002.safetensors:
88% 4.38G/4.98G [01:15<00:04, 140MB/s]odel-00001-of-00002.safetensors:
89% 4.45G/4.98G [01:15<00:03, 165MB/s]odel-00001-of-00002.safetensors:
91% 4.51G/4.98G [01:15<00:02, 182MB/s]odel-00001-of-00002.safetensors:
92% 4.58G/4.98G [01:16<00:02, 181MB/s]odel-00001-of-00002.safetensors:
93% 4.65G/4.98G [01:16<00:01, 215MB/s]odel-00001-of-00002.safetensors:
95% 4.72G/4.98G [01:19<00:05, 49.7MB/s]odel-00001-of-
00002.safetensors:  96% 4.78G/4.98G [01:20<00:03, 65.4MB/s]odel-00001-
of-00002.safetensors:  97% 4.85G/4.98G [01:20<00:01, 83.8MB/s]odel-
00001-of-00002.safetensors:  99% 4.92G/4.98G [01:26<00:02,
30.3MB/s]odel-00001-of-00002.safetensors: 100% 4.98G/4.98G
[01:26<00:00, 57.8MB/s]
Fetching 2 files: 100% 2/2 [01:26<00:00, 43.29s/it]
Loading checkpoint shards: 100% 2/2 [00:30<00:00, 15.17s/it]
generation_config.json: 100% 121/121 [00:00<00:00, 598kB/s]
 Model loaded successfully!
  Model device: cuda:0
  Model dtype: torch.float16
  GPU Memory: 2.97 GB allocated, 3.79 GB reserved

======================================================================
Starting evaluation on 10 subjects
======================================================================


Progress: 1/10 subjects

======================================================================
Evaluating subject: astronomy
======================================================================
Testing astronomy: 100% 152/152 [00:13<00:00, 11.24it/s]
 Result: 61/152 correct = 40.13%

Progress: 2/10 subjects

======================================================================
Evaluating subject: clinical_knowledge
======================================================================
Testing clinical_knowledge: 100% 265/265 [00:22<00:00, 11.76it/s]
 Result: 88/265 correct = 33.21%

Progress: 3/10 subjects

======================================================================
Evaluating subject: college_biology
======================================================================
Testing college_biology: 100% 144/144 [00:12<00:00, 11.77it/s]
 Result: 56/144 correct = 38.89%

Progress: 4/10 subjects

======================================================================
Evaluating subject: college_chemistry
======================================================================
Testing college_chemistry: 100% 100/100 [00:07<00:00, 13.91it/s]
 Result: 33/100 correct = 33.00%

Progress: 5/10 subjects

======================================================================
Evaluating subject: college_computer_science
======================================================================
Testing college_computer_science: 100% 100/100 [00:08<00:00,
11.33it/s]
 Result: 28/100 correct = 28.00%

Progress: 6/10 subjects

======================================================================
Evaluating subject: college_mathematics
======================================================================
Testing college_mathematics: 100% 100/100 [00:06<00:00, 15.42it/s]
 Result: 30/100 correct = 30.00%

Progress: 7/10 subjects

======================================================================
Evaluating subject: college_medicine
======================================================================
Testing college_medicine: 100% 173/173 [00:12<00:00, 13.59it/s]
```

```
 Result: 53/173 correct = 30.64%

Progress: 8/10 subjects

======================================================================
Evaluating subject: college_physics
======================================================================
Testing college_physics: 100% 102/102 [00:10<00:00, 10.15it/s]
 Result: 21/102 correct = 20.59%

Progress: 9/10 subjects

======================================================================
Evaluating subject: computer_security
======================================================================
Testing computer_security: 100% 100/100 [00:08<00:00, 11.58it/s]
 Result: 45/100 correct = 45.00%

Progress: 10/10 subjects

======================================================================
Evaluating subject: conceptual_physics
======================================================================
Testing conceptual_physics: 100% 235/235 [00:18<00:00, 13.04it/s]
 Result: 63/235 correct = 26.81%

======================================================================
EVALUATION SUMMARY
======================================================================
Model: allenai/OLMo-2-0425-1B
None (full precision)
Total Subjects: 10
Total Questions: 1471
Total Correct: 478
Overall Accuracy: 32.49%
Real Time: 2.3 minutes
CPU Time (process): 1.0 minutes
GPU Time (CUDA kernels): 1.9 minutes
Inferences timed: 1471
======================================================================

 Results saved to: llama_3.2_1b_mmlu_results_full_20260114_195647.json

ðŸ“Š Top 5 Subjects:
  1. computer_security: 45.00%
  2. astronomy: 40.13%
  3. college_biology: 38.89%
  4. clinical_knowledge: 33.21%
  5. college_chemistry: 33.00%
```

```
ðŸ“‰ Bottom 5 Subjects:
  1. college_medicine: 30.64%
  2. college_mathematics: 30.00%
  3. college_computer_science: 28.00%
  4. conceptual_physics: 26.81%
  5. college_physics: 20.59%

======================================================================
ðŸ’¾ To download results in Colab:
======================================================================
from google.colab import files
files.download('llama_3.2_1b_mmlu_results_full_20260114_195647.json')
Figure(1200x600)

ðŸ“· Accuracy plot saved to:
llama_3.2_1b_mmlu_accuracies_full_20260114_195647.png

âœ… Evaluation complete!
```

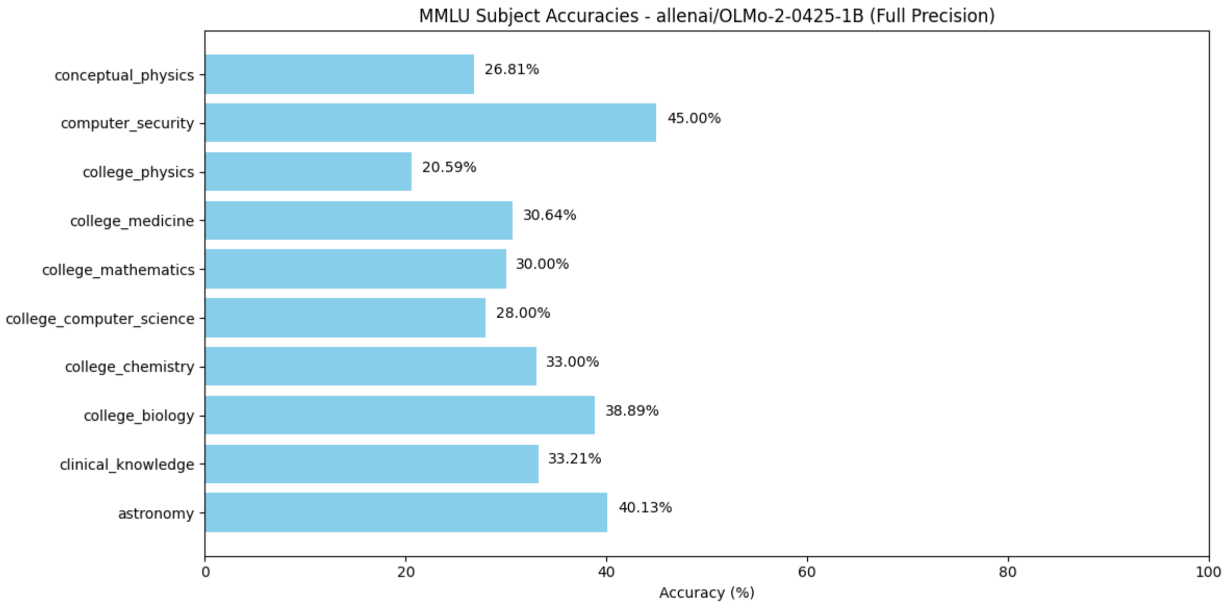
```
!python llama_mmlu_eval_quantized.py
```

```

======================================================================
Llama 3.2-1B MMLU Evaluation (Quantized)
======================================================================

======================================================================
Environment Check
======================================================================
 Running in Google Colab
 Platform: Linux (x86_64)
 GPU Available: Tesla T4
 GPU Memory: 15.83 GB
 Quantization disabled - loading full precision model
 Hugging Face authenticated

======================================================================
Configuration
======================================================================
Model: Qwen/Qwen2.5-0.5B
Device: cuda
Quantization: None (full precision)
Expected memory: ~2.5 GB (FP16)
Number of subjects: 10
======================================================================


Loading model Qwen/Qwen2.5-0.5B...
Device: cuda
tokenizer_config.json: 7.23kB [00:00, 17.6MB/s]
```

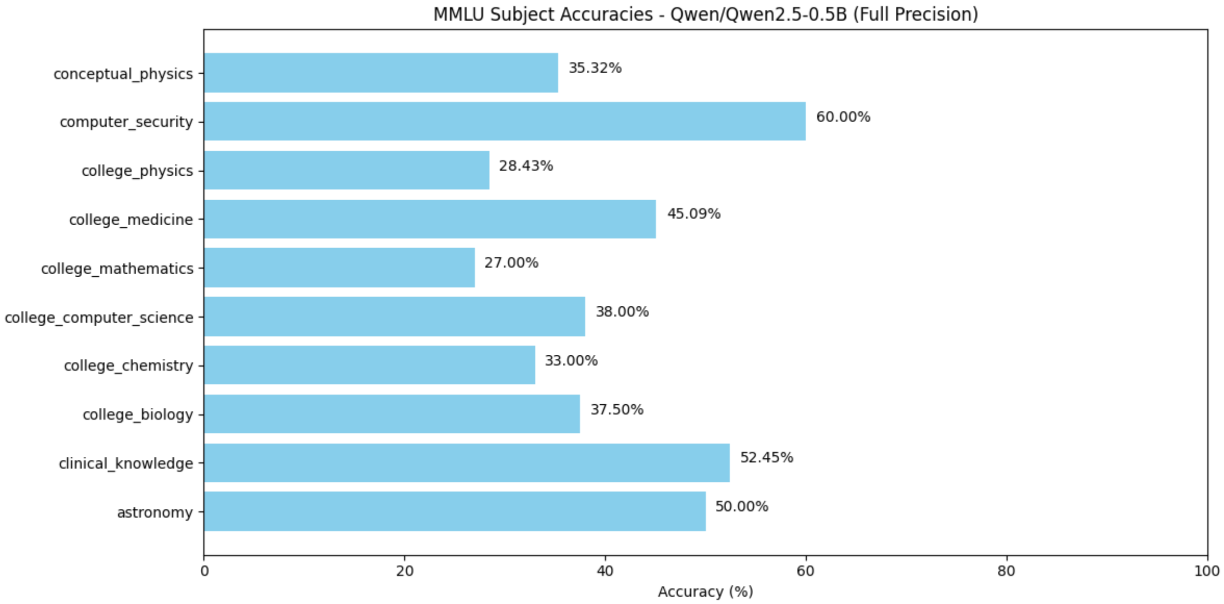
```
vocab.json: 2.78MB [00:00, 66.2MB/s]
merges.txt: 1.67MB [00:00, 8.83MB/s]
tokenizer.json: 7.03MB [00:00, 76.7MB/s]
 Tokenizer loaded
Loading model (this may take 2-3 minutes)...
config.json: 100% 681/681 [00:00<00:00, 3.99MB/s]
2026-01-14 19:47:09.820573: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to
register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
WARNING: All log messages before absl::InitializeLog() is called are
written to STDERR
E0000 00:00:1768420029.852633    5575 cuda_dnn.cc:8579] Unable to
register cuDNN factory: Attempting to register factory for plugin
cuDNN when one has already been registered
E0000 00:00:1768420029.862303    5575 cuda_blas.cc:1407] Unable to
register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
W0000 00:00:1768420029.885230    5575 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768420029.885258    5575 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768420029.885266    5575 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
W0000 00:00:1768420029.885273    5575 computation_placer.cc:177]
computation placer already registered. Please check linkage and avoid
linking the same target more than once.
2026-01-14 19:47:09.892102: I
tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow
binary is optimized to use available CPU instructions in performance-
critical operations.
To enable the following instructions: AVX2 FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
model.safetensors: 100% 988M/988M [00:14<00:00, 69.7MB/s]
generation_config.json: 100% 138/138 [00:00<00:00, 880kB/s]
 Model loaded successfully!
  Model device: cuda:0
  Model dtype: torch.float16
  GPU Memory: 1.00 GB allocated, 1.02 GB reserved

======================================================================
Starting evaluation on 10 subjects
======================================================================


Progress: 1/10 subjects
```

```
======================================================================
Evaluating subject: astronomy
======================================================================
Testing astronomy: 100% 152/152 [00:17<00:00,  8.84it/s]
 Result: 76/152 correct = 50.00%

Progress: 2/10 subjects

======================================================================
Evaluating subject: clinical_knowledge
======================================================================
Testing clinical_knowledge: 100% 265/265 [00:25<00:00, 10.24it/s]
 Result: 139/265 correct = 52.45%

Progress: 3/10 subjects

======================================================================
Evaluating subject: college_biology
======================================================================
Testing college_biology: 100% 144/144 [00:13<00:00, 10.72it/s]
 Result: 54/144 correct = 37.50%

Progress: 4/10 subjects

======================================================================
Evaluating subject: college_chemistry
======================================================================
Testing college_chemistry: 100% 100/100 [00:11<00:00,  8.68it/s]
 Result: 33/100 correct = 33.00%

Progress: 5/10 subjects

======================================================================
Evaluating subject: college_computer_science
======================================================================
Testing college_computer_science: 100% 100/100 [00:04<00:00,
22.76it/s]
 Result: 38/100 correct = 38.00%

Progress: 6/10 subjects

======================================================================
Evaluating subject: college_mathematics
======================================================================
Testing college_mathematics: 100% 100/100 [00:04<00:00, 23.35it/s]
 Result: 27/100 correct = 27.00%

Progress: 7/10 subjects

======================================================================
Evaluating subject: college_medicine
======================================================================
Testing college_medicine: 100% 173/173 [00:07<00:00, 22.40it/s]
 Result: 78/173 correct = 45.09%

Progress: 8/10 subjects

======================================================================
Evaluating subject: college_physics
======================================================================
Testing college_physics: 100% 102/102 [00:04<00:00, 23.92it/s]
 Result: 29/102 correct = 28.43%

Progress: 9/10 subjects

======================================================================
Evaluating subject: computer_security
======================================================================
Testing computer_security: 100% 100/100 [00:03<00:00, 25.67it/s]
 Result: 60/100 correct = 60.00%

Progress: 10/10 subjects

======================================================================
Evaluating subject: conceptual_physics
======================================================================
Testing conceptual_physics: 100% 235/235 [00:13<00:00, 17.29it/s]
 Result: 83/235 correct = 35.32%

======================================================================
EVALUATION SUMMARY
======================================================================
Model: Qwen/Qwen2.5-0.5B
None (full precision)
Total Subjects: 10
Total Questions: 1471
Total Correct: 617
Overall Accuracy: 41.94%
Real Time: 2.0 minutes
CPU Time (process): 1.2 minutes
GPU Time (CUDA kernels): 1.7 minutes
Inferences timed: 1471
======================================================================

 Results saved to: llama_3.2_1b_mmlu_results_full_20260114_194937.json

ðŸ“Š Top 5 Subjects:
  1. computer_security: 60.00%
  2. clinical_knowledge: 52.45%
  3. astronomy: 50.00%
  4. college_medicine: 45.09%
  5. college_computer_science: 38.00%

ðŸ“‰ Bottom 5 Subjects:
  1. college_biology: 37.50%
  2. conceptual_physics: 35.32%
  3. college_chemistry: 33.00%
  4. college_physics: 28.43%
  5. college_mathematics: 27.00%

======================================================================
ðŸ’¾ To download results in Colab:
======================================================================
from google.colab import files
files.download('llama_3.2_1b_mmlu_results_full_20260114_194937.json')
Figure(1200x600)

ðŸ“· Accuracy plot saved to:
llama_3.2_1b_mmlu_accuracies_full_20260114_194937.png

âœ… Evaluation complete!
```

```
from IPython.display import Image
Image(filename='/content/drive/MyDrive/Colab_Projects/RunningLMM/llama_3.2_1b_mmlu_accuracies_full_20260114_195647.png')
```



```
Image(filename='/content/drive/MyDrive/Colab_Projects/RunningLMM/
llama_3.2_1b_mmlu_accuracies_full_20260114_194937.png')
```


MMLU Subject Accuracies - Qwen/Qwen2.5-0.5B (Full Precision)
conceptual_physics
computer_security
college_physics
college_medicine
college_mathematics
college_computer_science
college_chemistry
college_biology
clinical_knowledge
astronomy
35.32%
60.00%
28.43%
45.09%
27.00%
38.00%
33.00%
37.50%
52.45%
50.00%
0
20
40
60
80
100
Accuracy (%)

```python
# -*- coding: utf-8 -*-
"""custom_chat.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1nqDm53GItlyoatCslqnFxXXpBguhwR6v
"""

!hf auth login

"""
Bare-Bones Chat Agent for Llama 3.2-1B-Instruct

This is a minimal chat interface that demonstrates:
1. How to load a model without quantization
2. How chat history is maintained and fed back to the model
3. The difference between plain text history and tokenized input

No classes, no fancy features - just the essentials.
"""

import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

# =============================================================================
# CONFIGURATION - Change these settings as needed
# =============================================================================

MODEL_NAME = "meta-llama/Llama-3.2-1B-Instruct"
CHAT_HISTORY_FLAG = True

# System prompt - This sets the chatbot's behavior and personality
# Change this to customize how the chatbot responds
SYSTEM_PROMPT = "You are a helpful AI assistant. Be concise and friendly."

# =============================================================================
# LOAD MODEL (NO QUANTIZATION)
# =============================================================================

print("Loading model (this takes 1-2 minutes)...")

# Load tokenizer (converts text to numbers and vice versa)
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)

# Load model in half precision (float16) for efficiency
# Use float16 on GPU, or float32 on CPU if needed
model = AutoModelForCausalLM.from_pretrained(
    MODEL_NAME,
    dtype=torch.float16,                        # Use FP16 for efficiency
    device_map="auto",                          # Automatically choose GPU/CPU
    low_cpu_mem_usage=True
)

model.eval()  # Set to evaluation mode (no training)
print(f"ГўЕ“вЂњ Model loaded! Using device: {model.device}")
print(f"ГўЕ“вЂњ Memory usage: ~2.5 GB (FP16)\n")

# =============================================================================
# CHAT HISTORY - This is stored as PLAIN TEXT (list of dictionaries)
# =============================================================================
# The chat history is a list of messages in this format:
# [
#   {"role": "system", "content": "You are a helpful assistant"},
#   {"role": "user", "content": "Hello!"},
#   {"role": "assistant", "content": "Hi! How can I help?"},
#   {"role": "user", "content": "What's 2+2?"},
#   {"role": "assistant", "content": "2+2 equals 4."}
# ]
#
# This is PLAIN TEXT - humans can read it
# The model CANNOT use this directly - it needs to be tokenized first

chat_history = []

# Add system prompt to history (this persists across the entire conversation)
chat_history.append({
    "role": "system",
    "content": SYSTEM_PROMPT
})

# =============================================================================
# CHAT LOOP
# =============================================================================

print("="*70)
print("Chat started! Type 'quit' or 'exit' to end the conversation.")
print("="*70 + "\n")

while True:
    # =========================================================================
    # STEP 1: Get user input (PLAIN TEXT)
    # =========================================================================
    user_input = input("You: ").strip()

    # Check for exit commands
    if user_input.lower() in ['quit', 'exit', 'q']:
        print("\nGoodbye!")
        break

    # Skip empty inputs
    if not user_input:
        continue


    # FLAG TO TURN OFF CHAT HISTORY
    if not CHAT_HISTORY_FLAG:
        chat_history = []

    # =========================================================================
    # STEP 2: Add user message to chat history (PLAIN TEXT)
    # =========================================================================
    # The chat history grows with each exchange
    # We append the new user message to the existing history
    chat_history.append({
        "role": "user",
        "content": user_input
    })


    # SLIDING WINDOW TOKEN TRUNCATION ADDITION ---------------------------
    if chat_history > 10:
        chat_history = chat_history[-10:]
    # ------------------------------------------------------------------

    # At this point, chat_history looks like:
    # [
    #   {"role": "system", "content": "You are helpful..."},
    #   {"role": "user", "content": "Hello!"},
    #   {"role": "assistant", "content": "Hi!"},
    #   {"role": "user", "content": "What's 2+2?"},      ГўвЂ Вђ Just added
    # ]
    # This is still PLAIN TEXT

    # ========================================================================
    # STEP 3: Convert chat history to model input (TOKENIZATION)
    # ========================================================================
    # The model needs numbers (tokens), not text
    # apply_chat_template() does two things:
    #   1. Formats the chat history with special tokens (like <|start|>, <|end|>)
    #   2. Converts the formatted text into token IDs (numbers)

    # First, apply_chat_template formats the history and converts to tokens
    input_ids = tokenizer.apply_chat_template(
        chat_history,                    # Our PLAIN TEXT history
        add_generation_prompt=True,      # Add prompt for assistant's response
        return_tensors="pt"              # Return as PyTorch tensor (numbers)
    ).to(model.device)

    # Create attention mask (1 for all tokens since we have no padding)
    attention_mask = torch.ones_like(input_ids)

    # Now input_ids is TOKENIZED - it's a tensor of numbers like:
    # tensor([[128000, 128006, 9125, 128007, 271, 2675, 527, 264, ...]])
    # These numbers represent our entire conversation history

    # ========================================================================
    # STEP 4: Generate assistant response (MODEL INFERENCE)
    # ========================================================================
    # The model looks at the ENTIRE chat history (in tokenized form)
    # and generates a response

    print("Assistant: ", end="", flush=True)

    with torch.no_grad():  # Don't calculate gradients (we're not training)
        outputs = model.generate(
            input_ids,
            attention_mask=attention_mask,   # Explicitly pass attention mask
            max_new_tokens=512,              # Maximum length of response
            do_sample=True,                  # Use sampling for variety
            temperature=0.7,                 # Lower = more focused, higher = more random
            top_p=0.9,                       # Nucleus sampling
            pad_token_id=tokenizer.eos_token_id
        )

    # outputs contains: [original input tokens + new generated tokens]
    # We only want the NEW tokens (the assistant's response)

    # ========================================================================
    # STEP 5: Decode the response (DETOKENIZATION)
    # ========================================================================
    # Extract only the newly generated tokens
    new_tokens = outputs[0][input_ids.shape[1]:]

    # Convert tokens (numbers) back to text (PLAIN TEXT)
    assistant_response = tokenizer.decode(
        new_tokens,
        skip_special_tokens=True  # Remove special tokens like <|end|>
    )

    print(assistant_response)  # Display the response

    # ========================================================================
    # STEP 6: Add assistant response to chat history (PLAIN TEXT)
    # ========================================================================
    # This is crucial! We add the assistant's response to the history
    # so the model remembers what it said in future turns

    chat_history.append({
        "role": "assistant",
        "content": assistant_response
    })

    # Now chat_history has grown again:
    # [
    #   {"role": "system", "content": "You are helpful..."},
    #   {"role": "user", "content": "Hello!"},
    #   {"role": "assistant", "content": "Hi!"},
    #   {"role": "user", "content": "What's 2+2?"},
    #   {"role": "assistant", "content": "4"}              ГўвЂ Вђ Just added
    # ]

    # When the loop repeats:
    # - User enters new message
    # - We add it to chat_history
    # - We tokenize the ENTIRE history (including all previous exchanges)
    # - Model sees everything and generates response
    # - We add response to history
    # - Repeat...

    # This is how the chatbot "remembers" the conversation!
    # Each turn, we feed it the ENTIRE conversation history

    print()  # Blank line for readability

# ============================================================================
# SUMMARY OF HOW CHAT HISTORY WORKS
# ============================================================================
"""
PLAIN TEXT vs TOKENIZED:

1. PLAIN TEXT (chat_history):
   - Human-readable format
   - List of dictionaries: [{"role": "user", "content": "Hi"}, ...]
   - Stored in memory between turns
   - Gets longer with each message

2. TOKENIZED (input_ids):
   - Numbers (token IDs)
   - Created fresh each turn from chat_history
   - This is what the model actually "reads"
   - Example: [128000, 128006, 9125, 128007, ...]

PROCESS EACH TURN:
   User input (text)
   ГўвЂ вЂњ
   Add to chat_history (text)
   ГўвЂ вЂњ
   Tokenize entire chat_history (text ГўвЂ вЂ™ numbers)
   ГўвЂ вЂњ
   Model generates response (numbers)
   ГўвЂ вЂњ
   Decode response (numbers ГўвЂ вЂ™ text)
   ГўвЂ вЂњ
   Add response to chat_history (text)
   ГўвЂ вЂњ
   Loop back to start

WHY FEED ENTIRE HISTORY?
- The model has no memory between calls
- Each generation is independent
- To "remember" previous turns, we must include them in the input
- This is why context length matters - longer conversations = more tokens

WHAT HAPPENS AS CONVERSATION GROWS?
- chat_history gets longer (more messages)
- Tokenized input gets longer (more tokens)
- Eventually hits model's max context length (for Llama 3.2: 128K tokens)
- Then you need context management (truncation, summarization, etc.)
- But for this simple demo, we let it grow without limit
"""
```