

# Run nemo-megatron-gpt-5B model with NVIDIA NeMo

## Introduction

NVIDIA NeMo is a powerful toolkit designed for researchers working on various conversational AI tasks, including automatic speech recognition (ASR), text-to-speech synthesis (TTS), large language models (LLMs), and natural language processing (NLP). It aims to facilitate the reuse of existing code and pretrained models while enabling the creation of new conversational AI models. In this tutorial, we will explore NeMo's capabilities and learn how to use the Megatron-GPT 5B language model for language modeling tasks.

## Model and Software References:

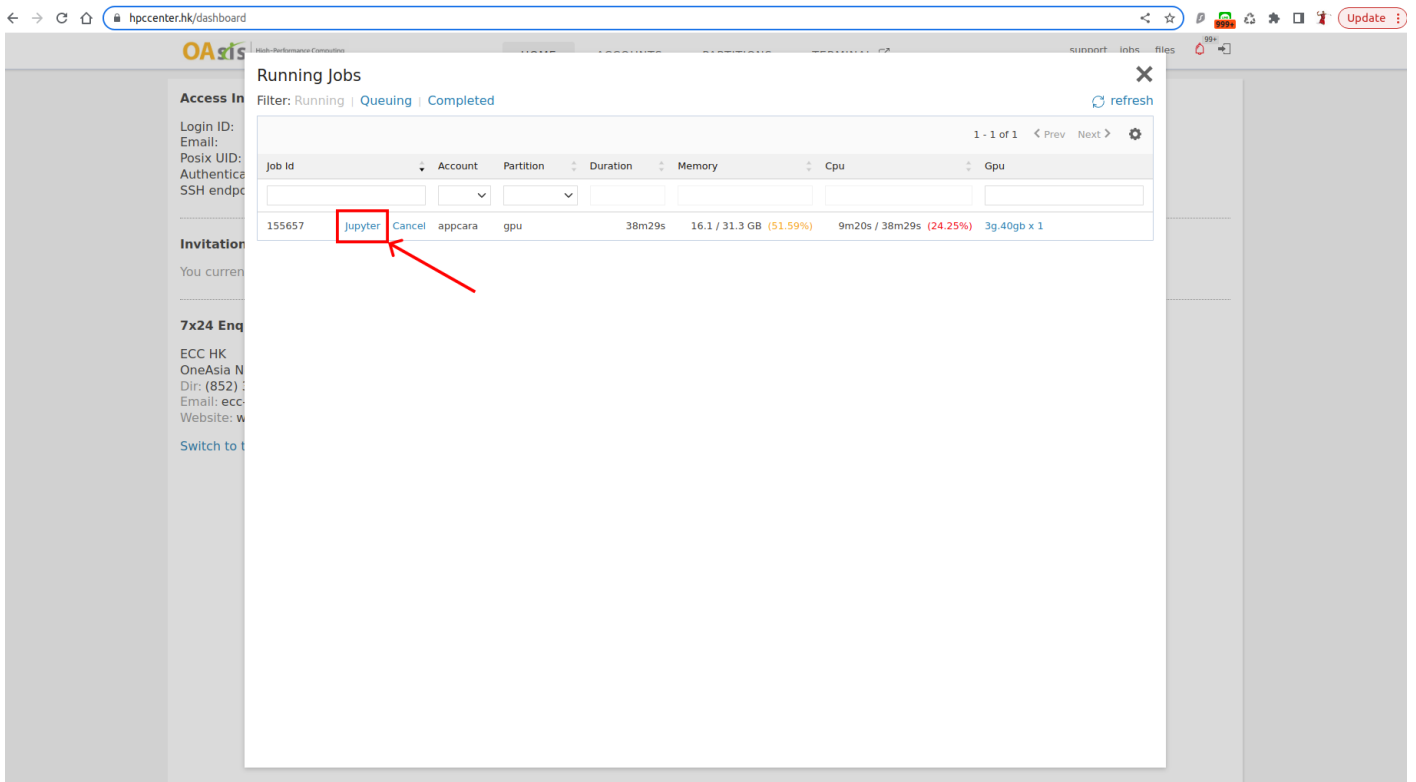
- NVIDIA NeMo: [<https://github.com/NVIDIA/NeMo>]
- nemo-megatron-gpt-5B: [<https://huggingface.co/nvidia/nemo-megatron-gpt-5B>]

## Launch Jupyter Lab Job

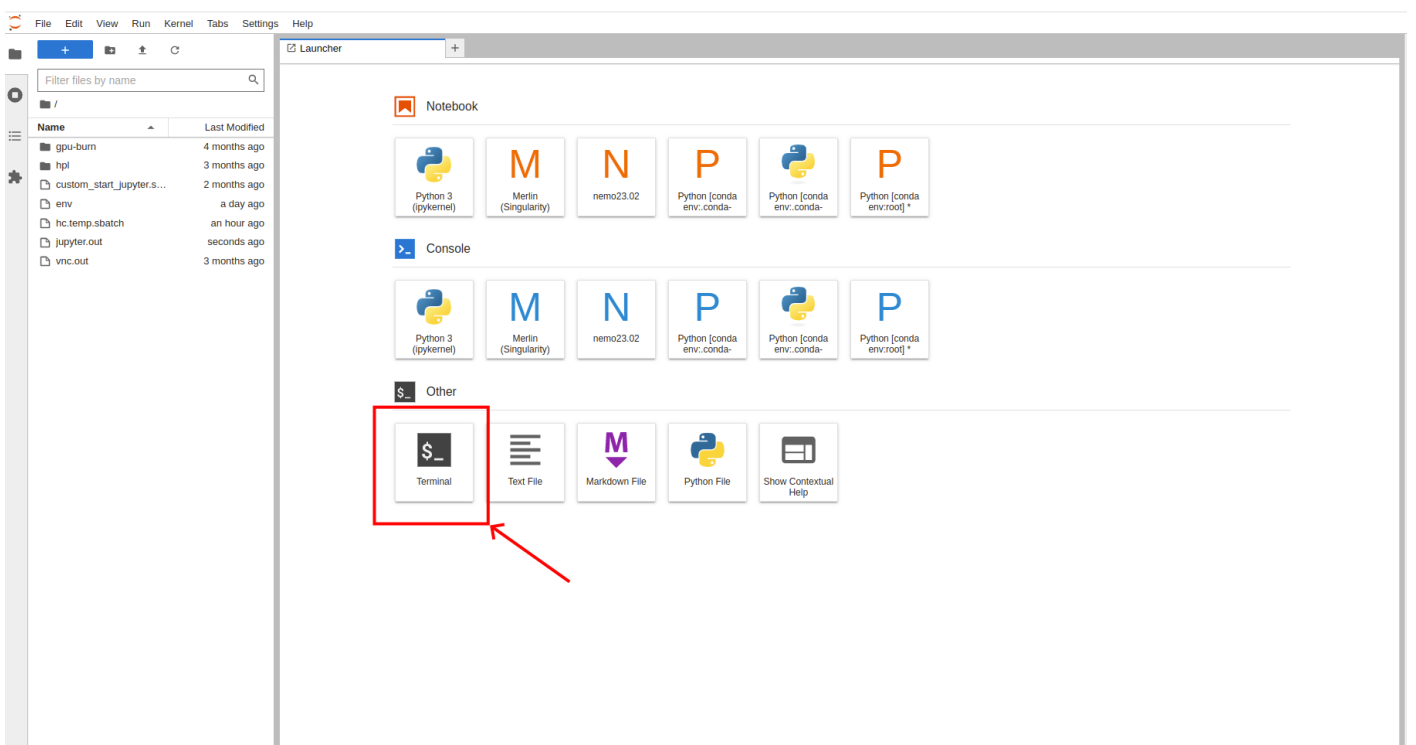
Create a Jupyter Lab job with the following specifications:

- CPU Cores: 4
- Memory: 64 GB
- GPU: 3g.40gb

Open your web browser and navigate to the Jupyter Lab web interface.



In the Jupyter Lab menu, open the Terminal.



## Enabling the NeMo Container Kernel in Jupyter Lab

Execute the following commands in the Terminal:

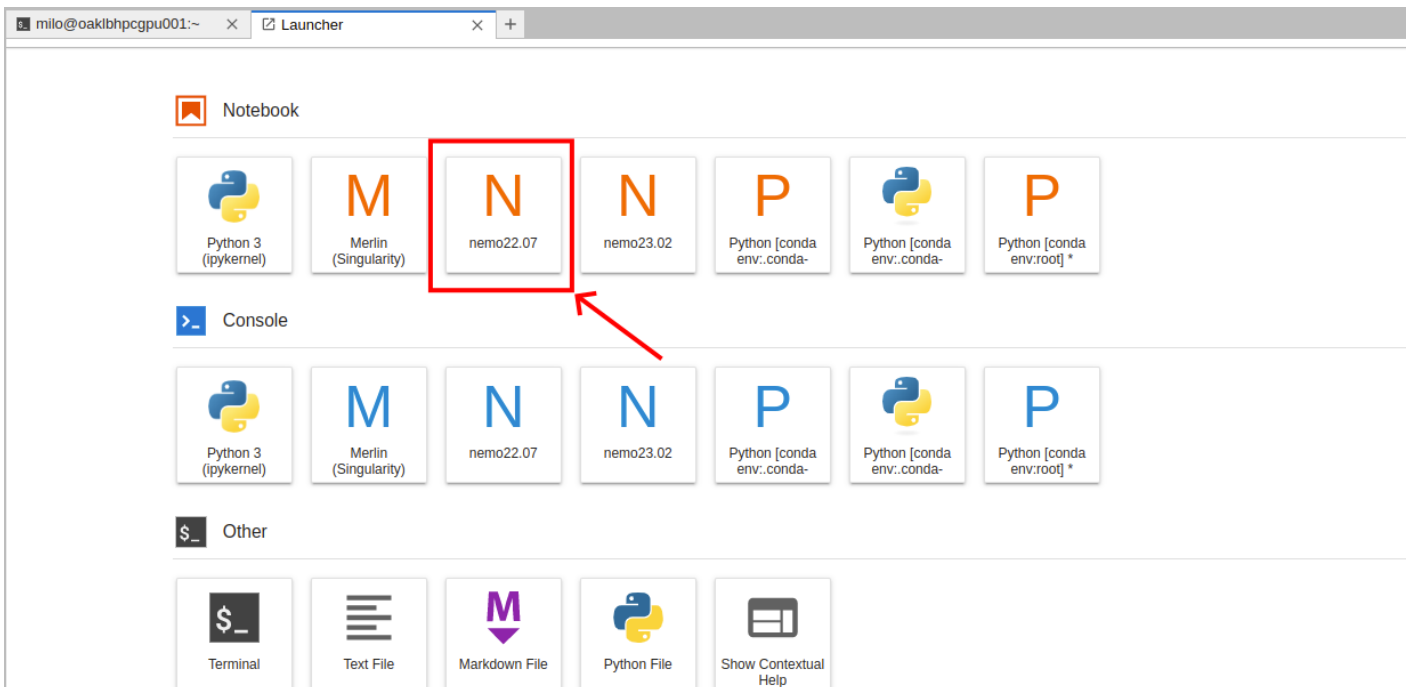
```
cd $HOME
mkdir -p .local/share/jupyter/kernels/ngc.nemo.22.07
```

```

echo '
{
  "language": "python",
  "argv": ["/usr/bin/singularity",
    "exec",
    "--nv",
    "-B",
    "/run/user:/run/user",
    "/pfss/containers/ngc.nemo.22.07.sif",
    "python",
    "-m",
    "ipykernel",
    "-f",
    "{connection_file}"
  ],
  "display_name": "nemo22.07"
}
' > .local/share/jupyter/kernels/ngc.nemo.22.07/kernel.json

```

After adding the content to the kernel.json file, refresh your browser by pressing F5. You should now see "nemo22.07" under the Notebook section in Jupyter Lab Launcher.



# Launch eval server

Execute the following command in the Terminal:

```
# set the TMPDIR environment variable
export TMPDIR=/pfss/scratch02/appcara/nlp/tmp

# start eval server with nemo-megatron-gpt-5B model by nemo container
singularity run --nv /pfss/containers/ngc.nemo.22.07.sif python
/pfss/scratch02/appcara/nlp/NeMo/examples/nlp/language_modeling/megatron_gpt_eval.py
gpt_model_file=/pfss/scratch02/appcara/nlp/nemo_gpt5B_fp16_tp1.nemo server=true
tensor_model_parallel_size=1 trainer.devices=1 port=5556
```

## Send prompts to the model

Copying the Jupyter Lab File:

```
# copy the jupyter example file into your home folder
cp $SCRATCH_APPCARA/nlp/nemo-megatron-gpt-template.ipynb $HOME
```

In the "File Browser" section of Jupyter Lab, locate the copied file and open it. Also change the kernel to **nemo22.07**.

The screenshot shows the Jupyter Lab interface. On the left is the 'File Browser' panel with a list of files. The file 'nemo-megatron-gpt-template.ipynb' is selected. The main panel shows the code editor with two cells. Cell [1] contains Python code for sending requests to the eval server. Cell [2] contains a print statement with a prompt. A 'Select Kernel' dialog box is open, showing a list of kernels. The kernel 'nemo22.07' is selected. The bottom status bar shows 'Mode: Command' and 'Ln 7, Col 18'.

File Browser:

Name	Last Modified
gou-burn	4 months ago
hpl	3 months ago
custom_start_jupyter.sbatch	2 months ago
env	a day ago
hc.temp.sbatch	11 minutes ago
jupyter.out	4 minutes ago
nemo-megatron-gpt-template.ipynb	4 minutes ago
vnc.out	3 months ago

Code Editor:

```
[1]: import json
import requests
port_num = 5556
headers = {'Content-Type': 'application/json'}
def request_data(data):
    resp = requests.put('http://localhost:{}/generate'.format(port_num),
                        data=json.dumps(data),
                        headers=headers)
    sentences = resp.json()['sentences'][0]
    return sentences

[2]: print(request_data({
    "sentences": ["Compose an engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions."],
    "tokens_to_generate": 2000,
    "temperature": 1.0,
    "add_BOS": False,
    "top_k": 0,
    "top_p": 0.9,
    "greedy": False,
    "all_probs": False,
    "repetition_penalty": 1.0,
    "min_tokens_to_generate": 10
}))
```

Select Kernel:

- nemo22.07
- Start Preferred Kernel
- Merlin (Singularity)
- nemo22.07
- nemo23.02
- Python [conda env: conda-chatbotDemo3]
- Python [conda env: conda-torch]
- Python [conda env: root]
- Python 3 (ipykernel)
- Use No Kernel
- No Kernel
- Use Kernel from Preferred Session
- nemo-megatron-gpt-template.ipynb
- Use Kernel from Other Session

Edit what you want to talk with chatbot in the "**sentences**" section of the file. Run the program.

The screenshot shows a JupyterLab environment. On the left is a file browser with a search bar and a list of files. The file `nemo-megatron-gpt-template.ipynb` is selected. The main area is a code editor with two tabs: `milo@oakthpcgpu001:~` and `nemo-megatron-gpt-template`. The active tab shows a Python script that defines a REST API endpoint `/generate`. The script takes a JSON request with a `data` field and returns a list of sentences generated by a GPT model. The output of the script is displayed below the code, showing a travel blog post about Hawaii. The blog post includes details about a recent trip, cultural experiences, and must-see attractions. It also mentions a flight to Honolulu and Hawaiian cuisine. The output is formatted as a JSON object with a `sentences` field.

```
[1]: import json
import requests
port_num = 5556
headers = {"Content-Type": "application/json"}
def request_data(data):
    resp = requests.put('http://localhost:{}'.format(port_num),
                        data=json.dumps(data),
                        headers=headers)
    sentences = resp.json()['sentences'][0]
    return sentences

[2]: print(request_data({
    "sentences": ["Compose an engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions."],
    "tokens_to_generate": 2000,
    "temperature": 1.0,
    "add_BOS": False,
    "top_k": 0,
    "top_p": 0.9,
    "greedy": False,
    "all_probs": False,
    "repetition_penalty": 1.2,
    "min_tokens_to_generate": 200,
}))
```

Compose an engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions.

The latter part of this sentence is not quite the same as "It should include". An alternative phrasing could be: "It's so important that you take some time off to see Hawaii. Do what makes sense for your situation (such as booking direct with The airline or booking through third party). Here are a couple of tips on how to make your trip extra special:"

A US / Canadian Airlines flight to Honolulu may be cheaper than one from the mainland US

Hawaiian cuisine will have increased in popularity compared to years past...

However, none of these sentences implies that a specific number of days/weeks/months were actually taken away for the trip. In fact, "a handful" is a perfectly fine amount of time which was enough for a positive experience without the use of any prepositions ("handful", "offload", etc.).

For vacationers visiting other countries in Australia who want to learn about Australia's natural wonders and history, the best way is by taking tours. During my journey I took several guided tours throughout Queensland and Sydney. And here are four examples where I felt lucky to receive fantastic service from professional guides... However, often guides can interpret their own tips incorrectly and some even seem rude. So what should I do? Which tour company provides high quality services? Or simply choose someone whom I can trust at all times - someone knowledgeable in a certain field but also willing to offer fair pricing policies as well? Do n't worry if you don't know anyone who sells Australian products. This article contains plenty of reviews on websites like TripAdvisor and GoodReads...

Note 1: It would help understand context considerably if the writer had provided proper citations. E.g., Biber et al 2016 and Lucas 2019 are relevant papers.

Note 2: One word was substituted for another in "transportation tools" and elsewhere for the sake of clarity, i.e. substitute to by its close relative to for instance when referring to experiences related to travel (i.e. it means learning about the country on road trips), rather than saying for example "car rental companies deliver good customer support as seen in customer reviews..."

A:

To write, I think I should utilize both Subject-Verb Agreement & Parallelism rules: those two have been together in writing since much earlier than two hundred years ago!

## Revision #3

Created 9 June 2023 04:14:47 by Milo Cheung

Updated 9 June 2023 06:43:42 by Milo Cheung