LLM supporting tools that make the ecosystem

Paul Rodriguez
San Diego Supercomputer Center

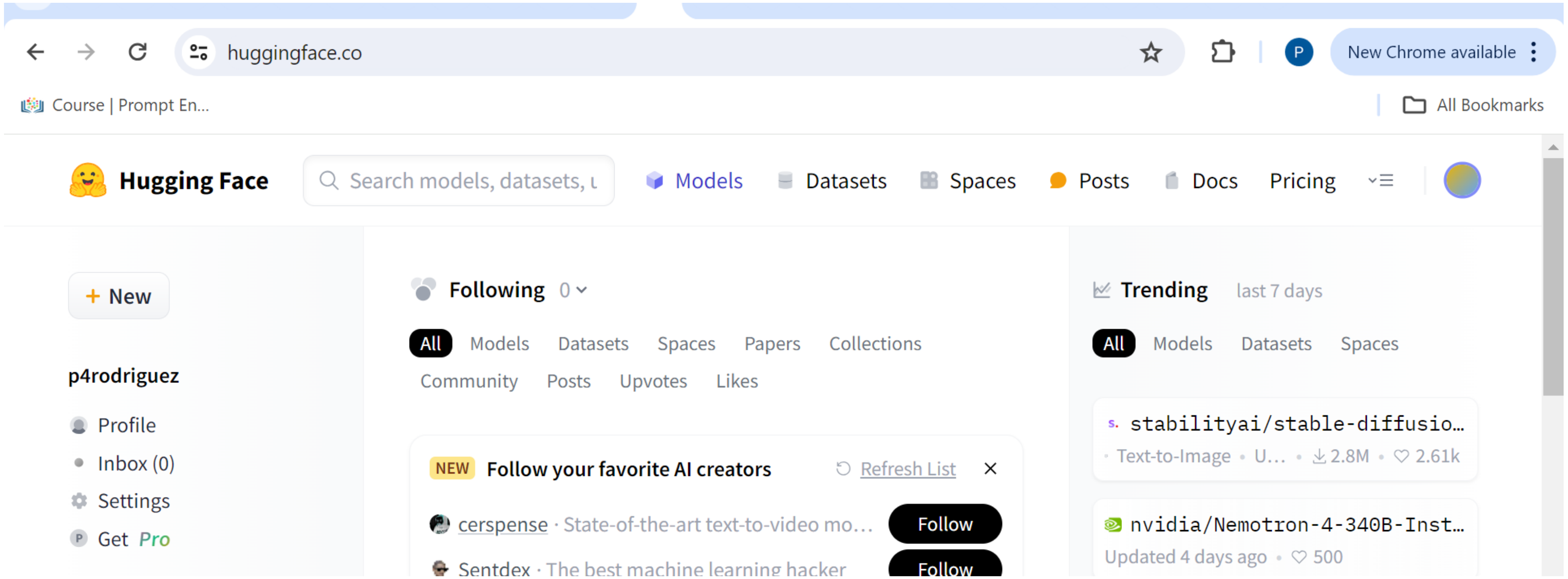
2024 CIML Summer Institute

Outline

- The Hugging Face ecosystem
- Langchain/Langgraph
- Using Hugging Face tutorials discussion/demo

Hugging Face Hub

- huggingface.com is a hub of models, data, tutorials for using AI models



cerspense · State-of-the-art text-to-video mo…

Sentdex · The best machine learning hacker

stabilityai/stable-diffusio…
Text-to-Image · U… · 2.8M · 2.61k

nvidia/Nemotron-4-340B-Inst…
Updated 4 days ago · 500

Using Hugging Face (HF)

- HF provides python packages to make it (relatively) easy to run models

- Accessing models and/or data requires an HF authentification token

- Some of the main packages are:

pipeline:	to run inferencing
diffusers:	for diffusion models
transformers:	for LLMs
accelerate:	for efficient and/or parallel execution
datasets:	to access data from hub

Hugging Face Abstractions

- HF package are intended to be more abstract than a Google, Meta, Msoft, etc. package, and work the same for all of them

- For example, the pipeline function is built on more basic functions:

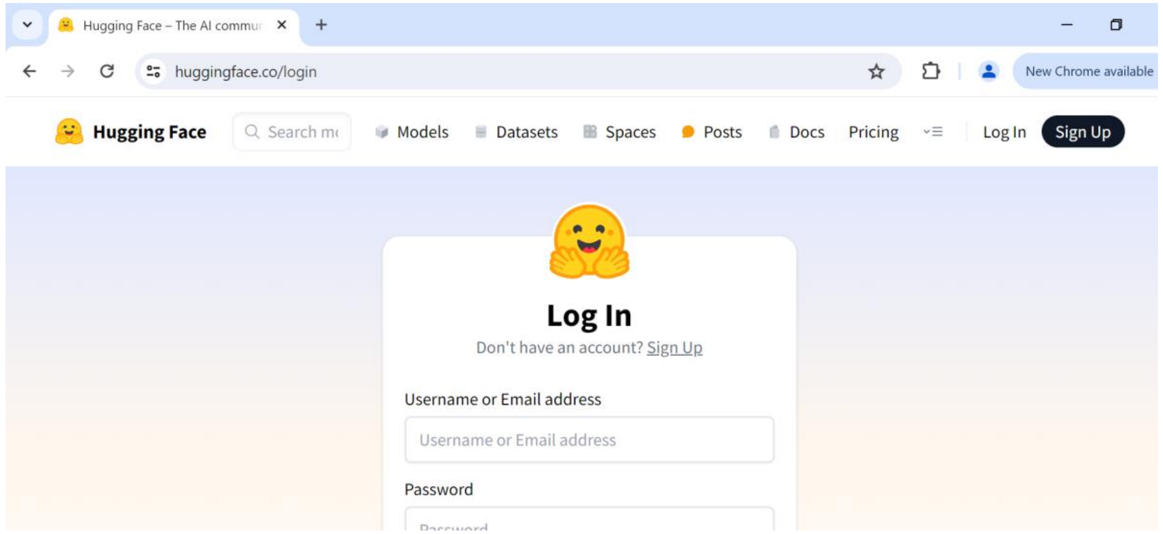
get “config”	(to get the model architecture)
get “model”	(to download a pretrained model)
get “tokenizer”	(to get the appropriate tokenizer)

- These can be invoked more directly if you want to do a pre-training or fine-tuning implementation, or experimentation

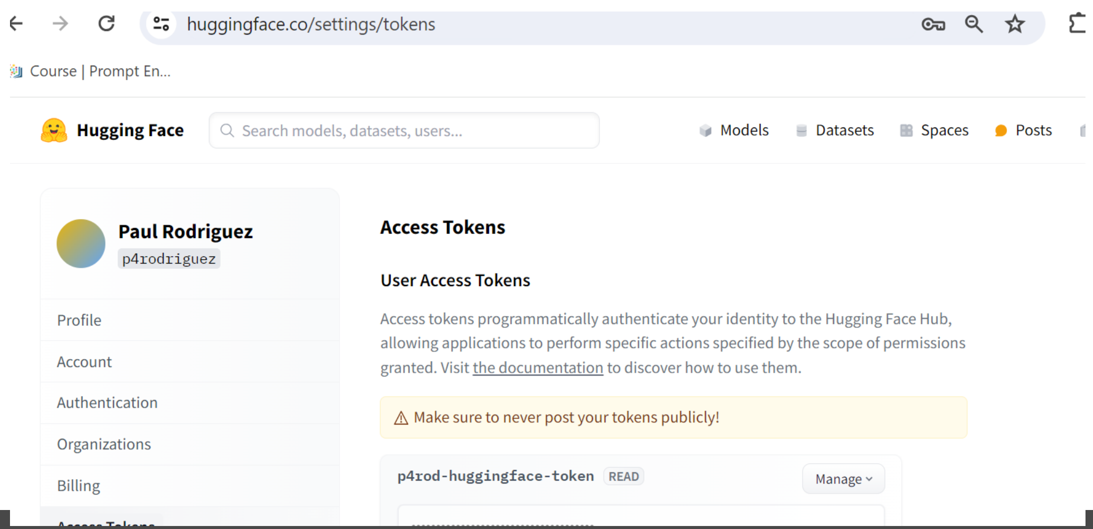
Getting account set up

- Create an account on huggingface and get authentication token

huggingface.co/login



Aacount -> settings -> get token



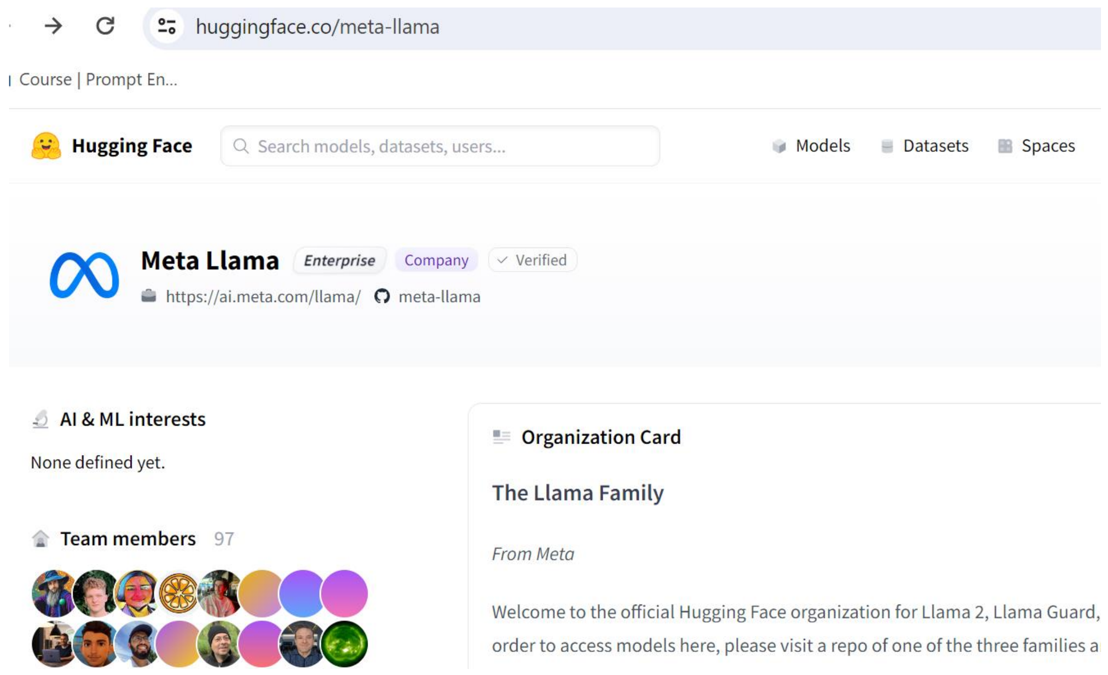
Access tokens programmatically authenticate your identity to the Hugging Face Hub, allowing applications to perform specific actions specified by the scope of permissions granted. Visit the documentation to discover how to use them.

⚠ Make sure to never post your tokens publicly!

p4rod-huggingface-token READ Manage

Hugging Face model repo

- **Example: Meta has a family of Llama models that vary by size, response training, ‘hf’ format, release date, etc..**



huggingface.co/meta-llama

Course | Prompt En...

Hugging Face — Search models, datasets, users... — Models — Datasets — Spaces

Meta Llama Enterprise Company Verified

https://ai.meta.com/llama/ meta-llama

AI & ML interests

None defined yet.

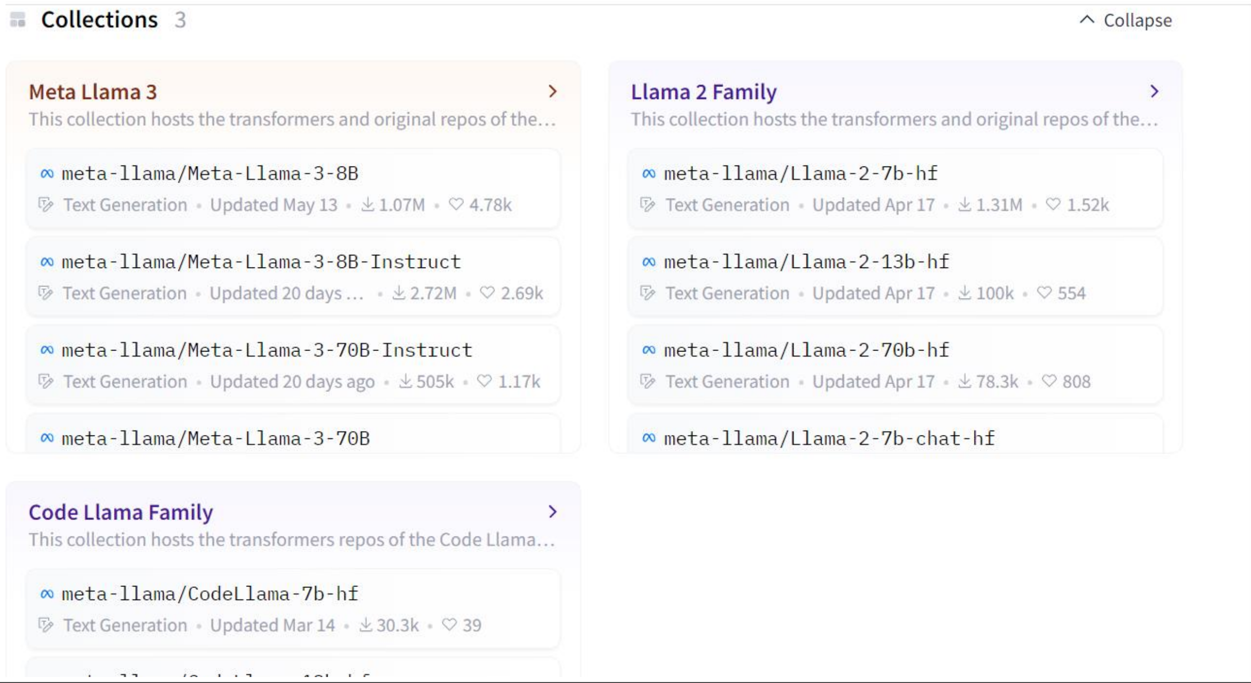
Team members 97

Organization Card

The Llama Family

From Meta

Welcome to the official Hugging Face organization for Llama 2, Llama Guard, order to access models here, please visit a repo of one of the three families a



Collections 3 — Collapse

Meta Llama 3
This collection hosts the transformers and original repos of the...

- meta-llama/Meta-Llama-3-8B — Text Generation • Updated May 13 • 1.07M • 4.78k
- meta-llama/Meta-Llama-3-8B-Instruct — Text Generation • Updated 20 days ... • 2.72M • 2.69k
- meta-llama/Meta-Llama-3-70B-Instruct — Text Generation • Updated 20 days ago • 505k • 1.17k
- meta-llama/Meta-Llama-3-70B

Llama 2 Family
This collection hosts the transformers and original repos of the...

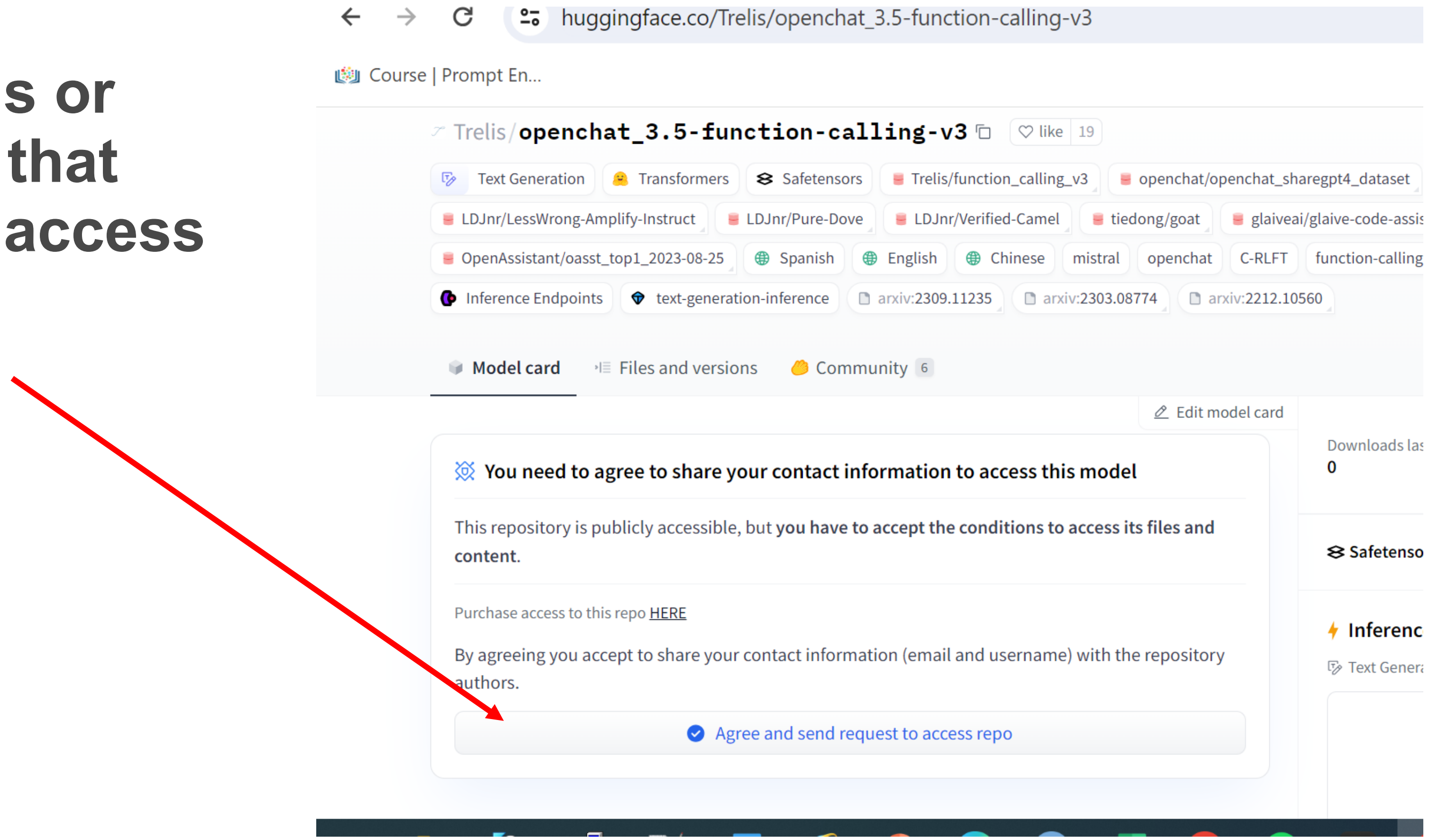
- meta-llama/Llama-2-7b-hf — Text Generation • Updated Apr 17 • 1.31M • 1.52k
- meta-llama/Llama-2-13b-hf — Text Generation • Updated Apr 17 • 100k • 554
- meta-llama/Llama-2-70b-hf — Text Generation • Updated Apr 17 • 78.3k • 808
- meta-llama/Llama-2-7b-chat-hf

Code Llama Family
This collection hosts the transformers repos of the Code Llama...

- meta-llama/CodeLlama-7b-hf — Text Generation • Updated Mar 14 • 30.3k • 39

Hugging Face model repo

- Some models or data require that you request access here



huggingface.co/Trelis/openchat_3.5-function-calling-v3

Course | Prompt En...

Trelis / openchat_3.5-function-calling-v3 like 19

Text Generation · Transformers · Safetensors · Trelis/function_calling_v3 · openchat/openchat_sharegpt4_dataset · LDJnr/LessWrong-Amplify-Instruct · LDJnr/Pure-Dove · LDJnr/Verified-Camel · tiedong/goat · glaiveai/glaive-code-assis · OpenAssistant/oasst_top1_2023-08-25 · Spanish · English · Chinese · mistral · openchat · C-RLFT · function-calling · Inference Endpoints · text-generation-inference · arxiv:2309.11235 · arxiv:2303.08774 · arxiv:2212.10560

Model card · Files and versions · Community 6

Edit model card

You need to agree to share your contact information to access this model

This repository is publicly accessible, but **you have to accept the conditions to access its files and content.**

Purchase access to this repo HERE

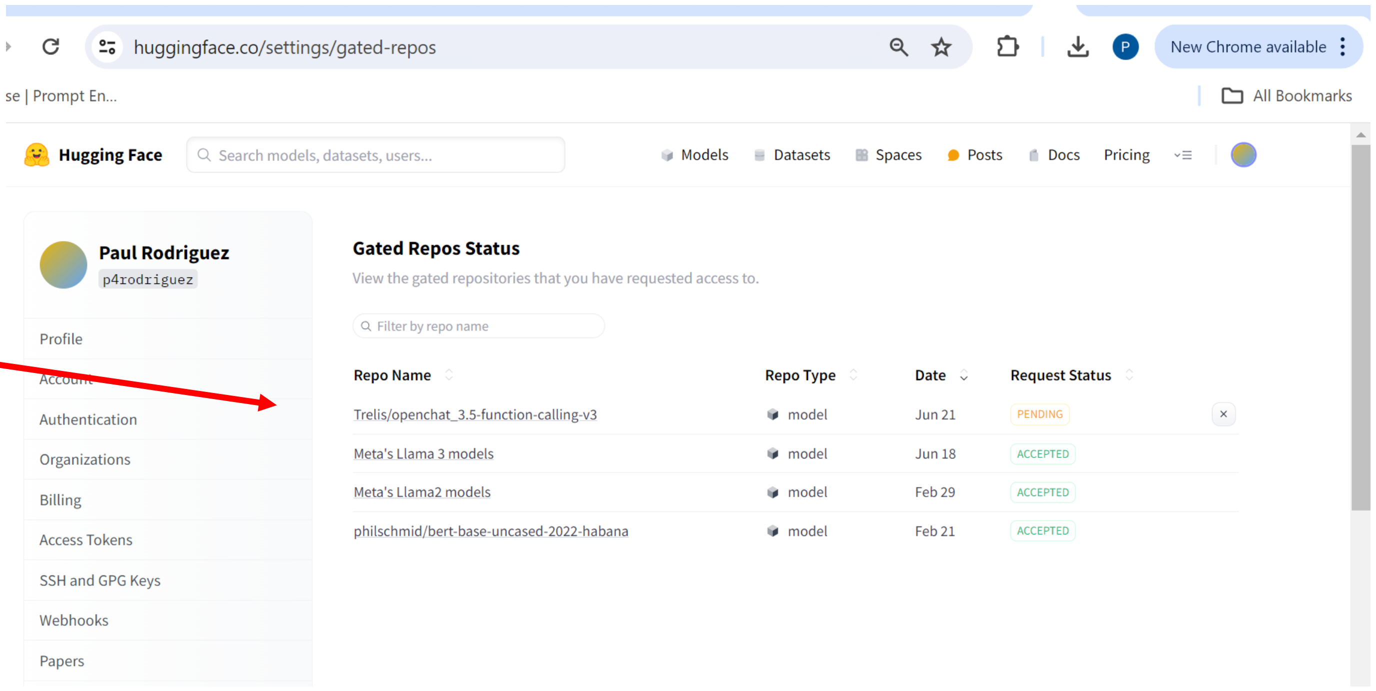
By agreeing you accept to share your contact information (email and username) with the repository authors.

Agree and send request to access repo

Hugging Face model repo

- Some models or data require that you request access

- Then go to
  account->
  setting->
  gated repo



huggingface.co/settings/gated-repos

Paul Rodriguez
p4rodriguez

Profile
Account
Authentication
Organizations
Billing
Access Tokens
SSH and GPG Keys
Webhooks
Papers

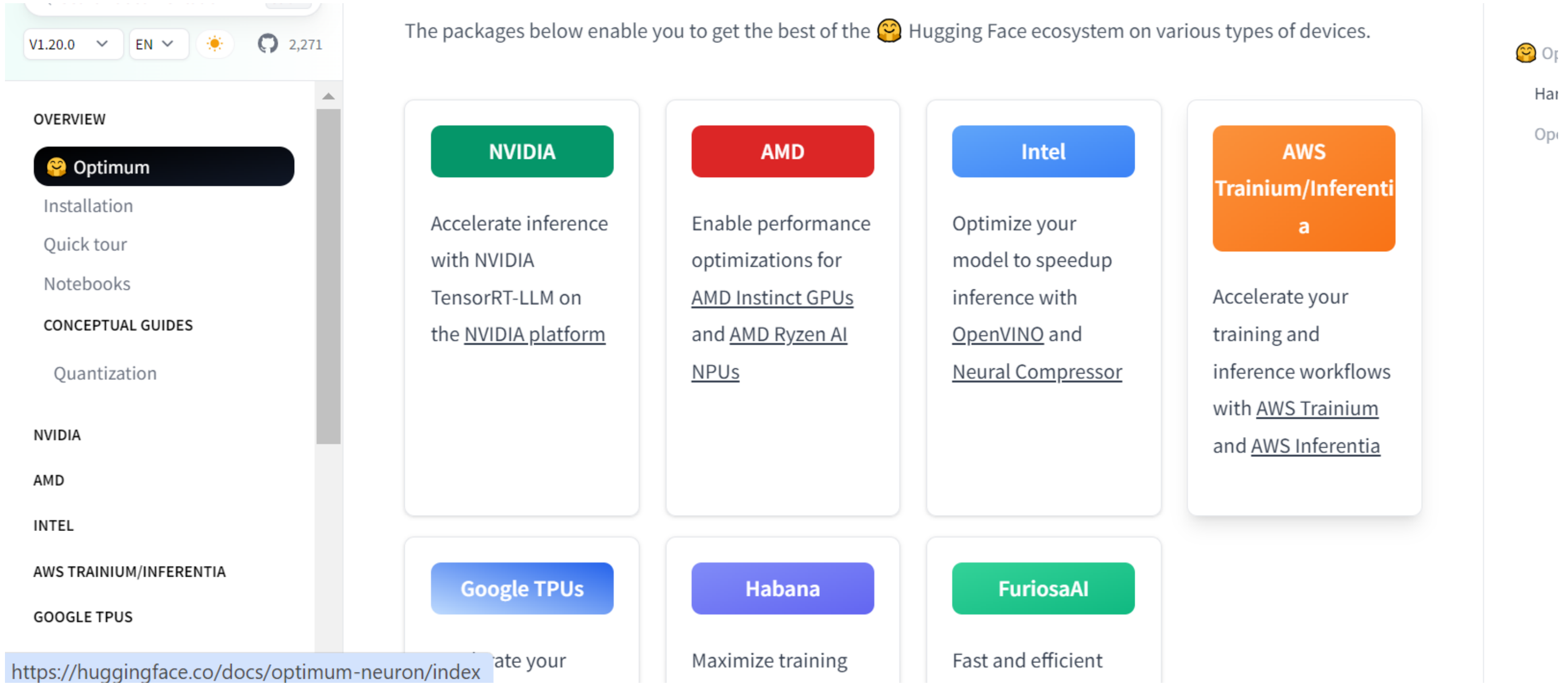
Gated Repos Status

View the gated repositories that you have requested access to.

Repo Name	Repo Type	Date	Request Status
Trelis/openchat_3.5-function-calling-v3	model	Jun 21	PENDING
Meta's Llama 3 models	model	Jun 18	ACCEPTED
Meta's Llama2 models	model	Feb 29	ACCEPTED
philschmid/bert-base-uncased-2022-habana	model	Feb 21	ACCEPTED

Hardware Partners

- Hugging Face also incorporates accelerator libraries into their packages



The packages below enable you to get the best of the 🤗 Hugging Face ecosystem on various types of devices.

NVIDIA

Accelerate inference with NVIDIA TensorRT-LLM on the NVIDIA platform

AMD

Enable performance optimizations for AMD Instinct GPUs and AMD Ryzen AI NPUs

Intel

Optimize your model to speedup inference with OpenVINO and Neural Compressor

AWS Trainium/Inferentia

Accelerate your training and inference workflows with AWS Trainium and AWS Inferentia

Google TPUs

ate your

Habana

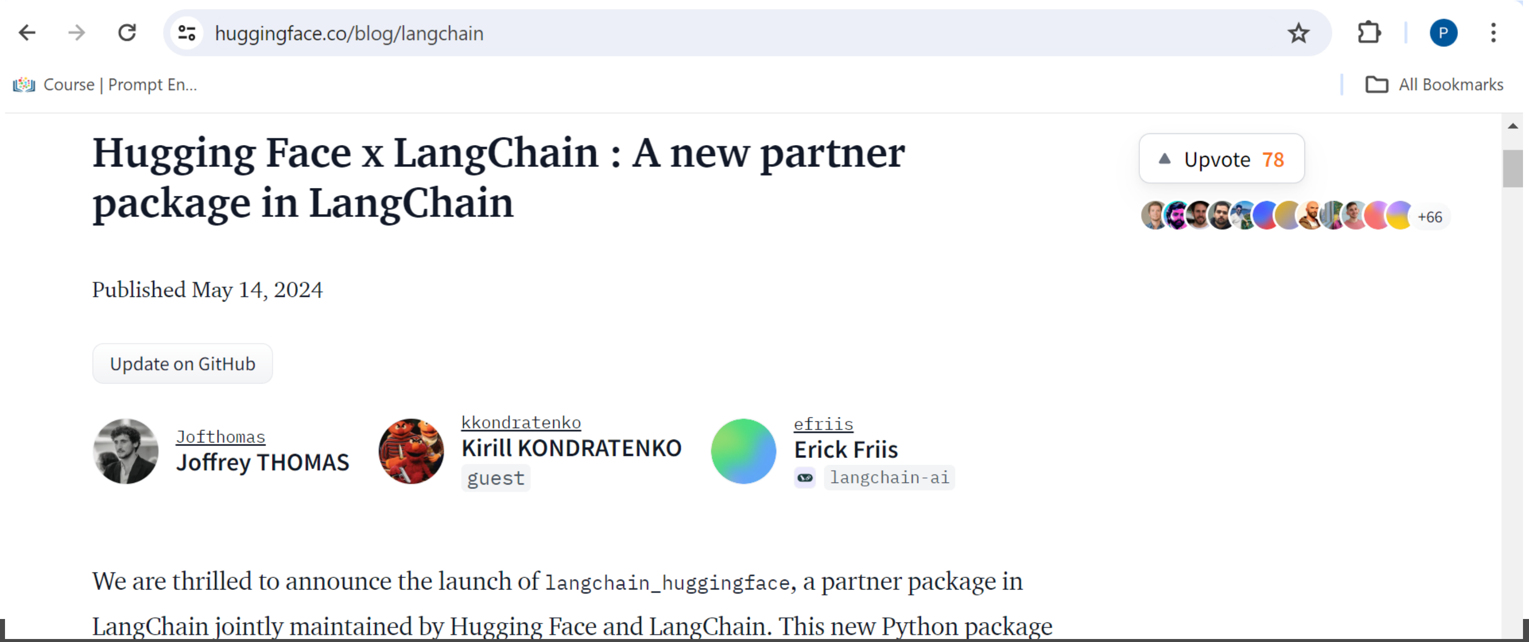
Maximize training

FuriosaAI

Fast and efficient

Partner packages

- **Hugging Face incorporates other packages**
- **For example, Langchain helps run RAG applications by:**
  - accessing PDF files or URL text as raw documents
  - creating database of documents (split into chunks and vectorized)
  - Setting up prompts that include relevant contexts



Hugging Face x LangChain : A new partner package in LangChain

Published May 14, 2024

Jofthomas
Joffrey THOMAS

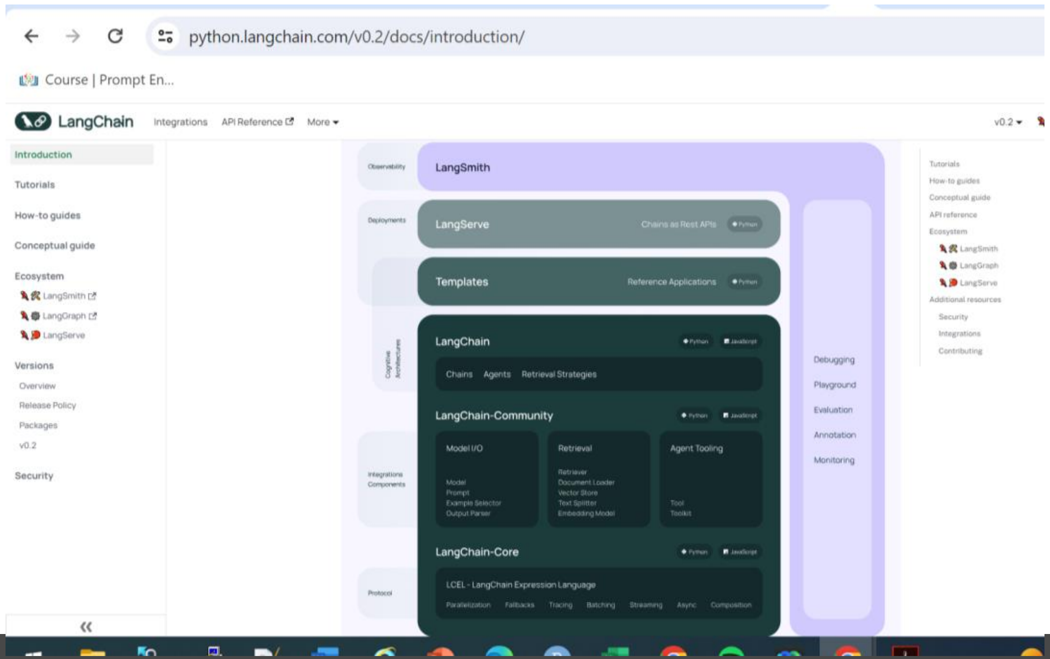
kkondratenko
Kirill KONDRATENKO
guest

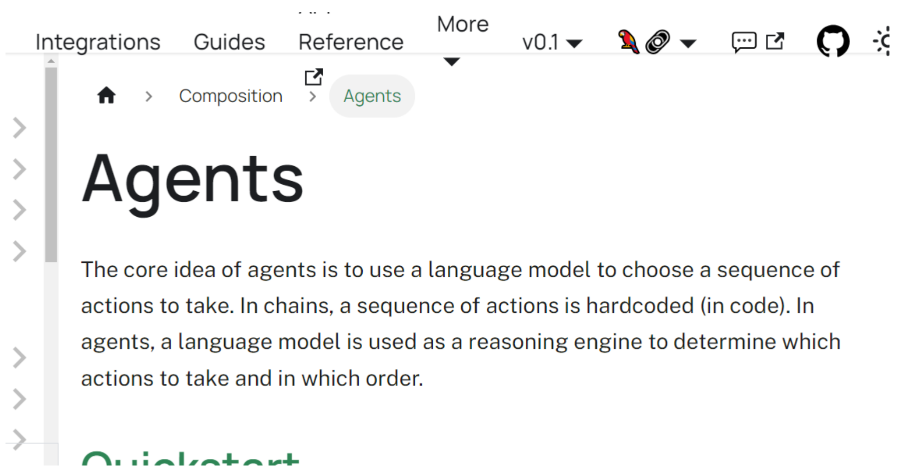
efriis
Erick Friis
langchain-ai

We are thrilled to announce the launch of `langchain_huggingface`, a partner package in LangChain jointly maintained by Hugging Face and LangChain. This new Python package

LangChain/LangGraph:

- **Not an LLM provider (like openAI, or Hugging Face) but provides a standard interface**
- **Has ecosystem of tools beyond RAG – e.g. Agents:** "a language model is used as a reasoning engine to determine which actions to take and in which order."


python.langchain.com/v0.2/docs/introduction/



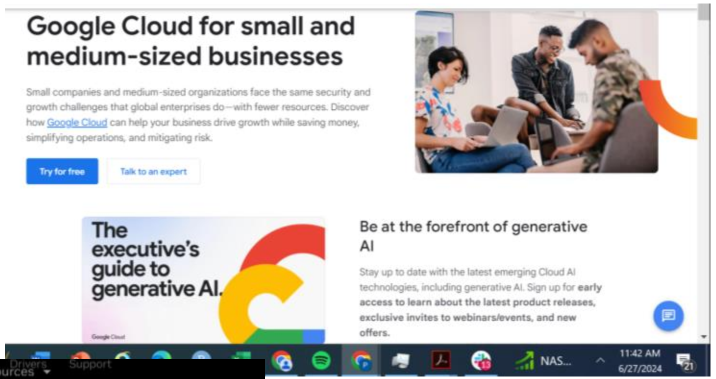
Agents

The core idea of agents is to use a language model to choose a sequence of actions to take. In chains, a sequence of actions is hardcoded (in code). In agents, a language model is used as a reasoning engine to determine which actions to take and in which order.

Other ecosystems to run LLMs

Many Big Tech firms have LLM services that provide 'turn key' solutions/tools for RAG, with APIs, open models, interfaces, deployment., etc.. geared for businesses.

- **Nvidia (NeMo)**
- **Google (cloud) AI**
- **LlamaIndex (meta)**
- **Etc…**

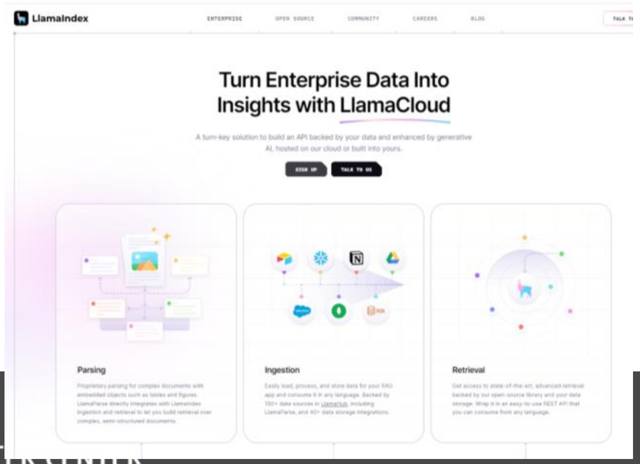


Google Cloud for small and medium-sized businesses

Small companies and medium-sized organizations face the same security and growth challenges that global enterprises do—with fewer resources. Discover how Google Cloud can help your business drive growth while saving money, simplifying operations, and mitigating risk.

Be at the forefront of generative AI

Stay up to date with the latest emerging Cloud AI technologies, including generative AI. Sign up for **early access to learn about the latest product releases, exclusive invites to webinars/events, and new offers.**



Turn Enterprise Data Into Insights with LlamaCloud

A turn-key solution to build an API backed by your data and enhanced by generative AI, hosted on our cloud or built into yours.

Parsing

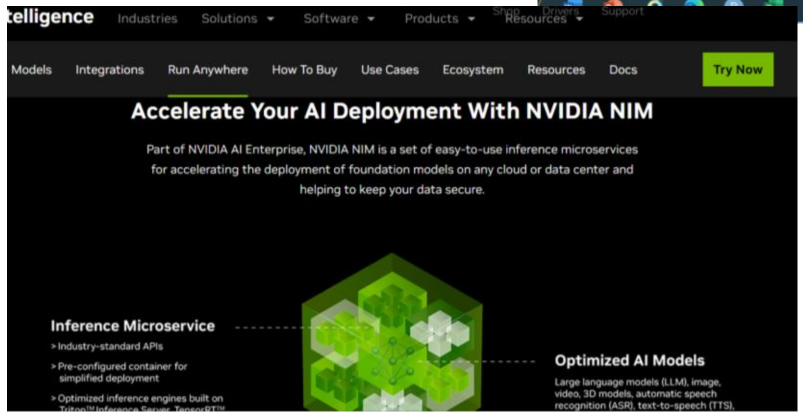
Proprietary parsing for complex documents with embedded objects such as tables and figures. LlamaParse directly integrates with LlamaIndex ingestion and retrieval to let you build retrieval over complex, semi-structured documents.

Ingestion

Easily load, process, and store data for your RAG app and consume it in any language. Backed by 150+ data sources in LlamaHub, including LlamaParse, and 40+ data storage integrations.

Retrieval

Get access to state-of-the-art, advanced retrieval backed by our open-source library and your data storage. Wrap it in an easy-to-use REST API that you can consume from any language.



Accelerate Your AI Deployment With NVIDIA NIM

Part of NVIDIA AI Enterprise, NVIDIA NIM is a set of easy-to-use inference microservices for accelerating the deployment of foundation models on any cloud or data center and helping to keep your data secure.

Inference Microservice

- Industry-standard APIs
- Pre-configured container for simplified deployment
- Optimized inference engines built on Triton™ Inference Server, TensorRT™

Optimized AI Models

Large language models (LLM), image, video, 3D models, automatic speech recognition (ASR), text-to-speech (TTS),

Keras extension for NLP

- **KerasNLP also has models, data, functions to run different LLMs (no tokens)**

- **Includes pre-trained LLMs base or full model, for example:**

• GPT2Backbone	the model without task specific output layers
• GPT2CausalLM	the model with output predictions
• GPT2CausalLMPreprocessor	the preprocessor that feeds model.fit

Expanse Demo/Exercise setting up HugginFace and Langchain

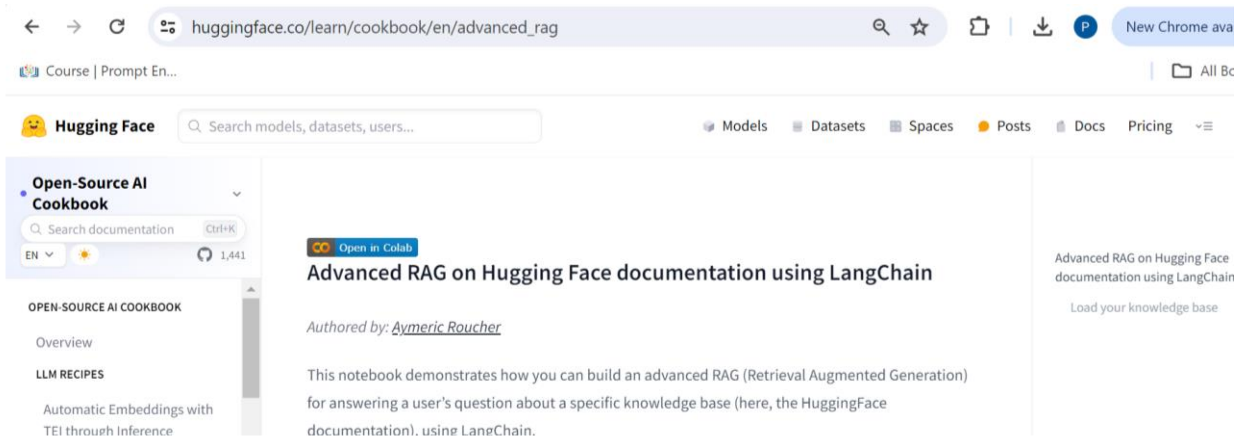
Case study: Jupyter notebook LLM tutorials (with pip install) on Expanse

- Start with Huggingface/Langchain for RAG

https://huggingface.co/learn/cookbook/en/advanced_rag

- But use a URL about SLURM for the raw documents

https://python.langchain.com/v0.2/docs/integrations/document_loaders/url/


huggingface.co/learn/cookbook/en/advanced_rag

Course | Prompt En...

Hugging Face | Search models, datasets, users... | Models | Datasets | Spaces | Posts | Docs | Pricing

Open-Source AI Cookbook

Search documentation Ctrl+K

EN | 1,441

OPEN-SOURCE AI COOKBOOK

Overview

LLM RECIPES

Automatic Embeddings with TEI through Inference

Open in Colab

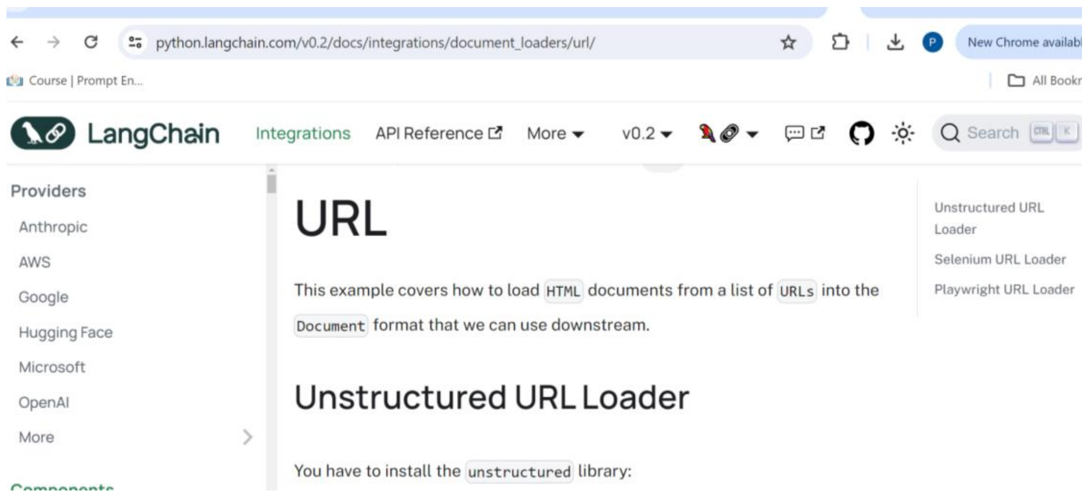
Advanced RAG on Hugging Face documentation using LangChain

Authored by: Aymeric Roucher

This notebook demonstrates how you can build an advanced RAG (Retrieval Augmented Generation) for answering a user's question about a specific knowledge base (here, the HuggingFace documentation), using LangChain.

Advanced RAG on Hugging Face documentation using LangChain

Load your knowledge base


python.langchain.com/v0.2/docs/integrations/document_loaders/url/

Course | Prompt En...

LangChain | Integrations | API Reference | More | v0.2 | Search

Providers

Anthropic

AWS

Google

Hugging Face

Microsoft

OpenAI

More

URL

This example covers how to load `HTML` documents from a list of `URLs` into the `Document` format that we can use downstream.

Unstructured URL Loader

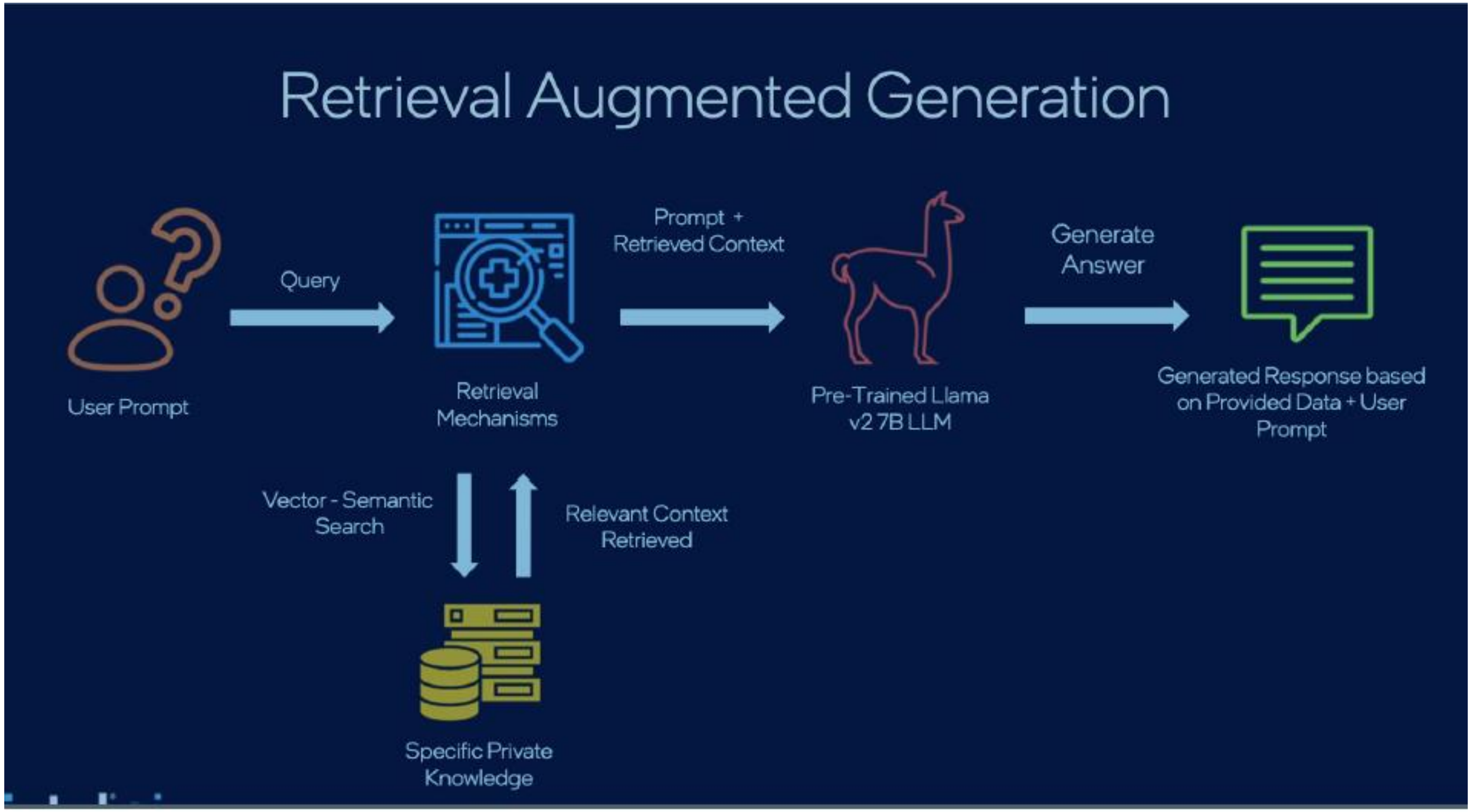
You have to install the `unstructured` library:

Unstructured URL Loader

Selenium URL Loader

Playwright URL Loader

RAG overview (from the Mai's slide)


Retrieval Augmented Generation
User Prompt
Query
Retrieval Mechanisms
Prompt + Retrieved Context
Pre-Trained Llama v2 7B LLM
Generate Answer
Generated Response based on Provided Data + User Prompt
Vector - Semantic Search
Relevant Context Retrieved
Specific Private Knowledge

Running notebook tutorials on Expanse

- On Expanse we can run a python singularity container that starts up Jupyter notebooks on a GPU and run !pip install commands

    Pro: easy to follow instructions

    Con: some things not scalable for long workflows

Running notebook tutorials on Expanse

- On Expanse we can run a python singularity container that starts up Jupyter notebooks on a GPU and run !pip install commands

  Pro: easy to follow instructions

  Con: some things not scalable for long workflows

OR

- On Expanse we can get a GPU node and use the command line to run pip install commands

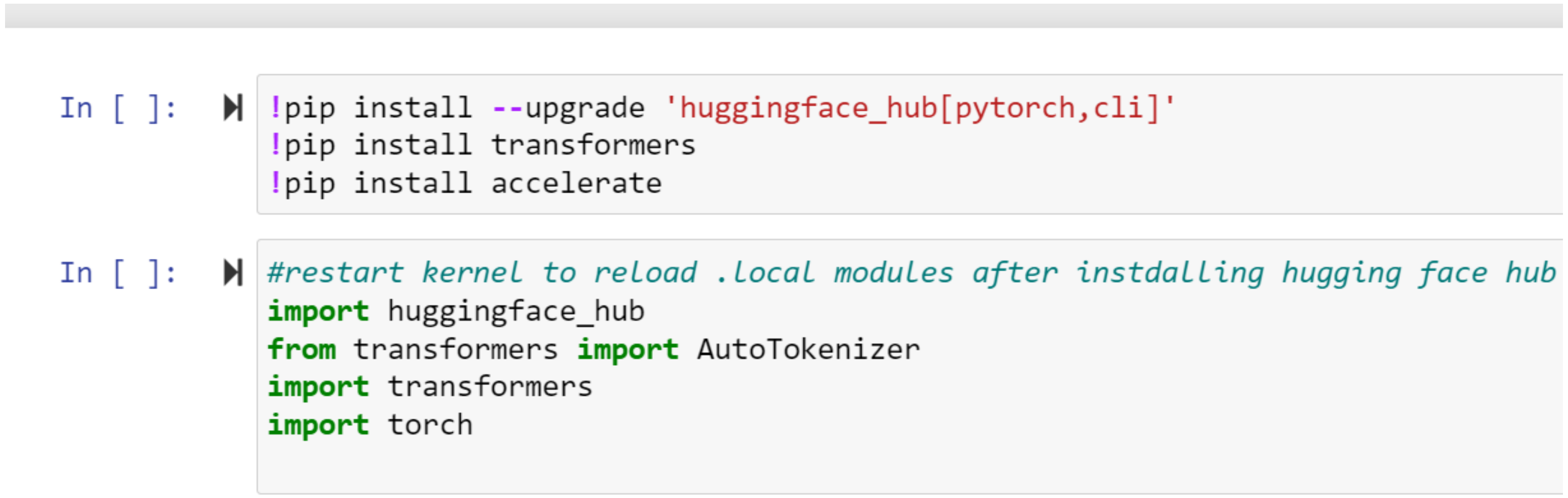
  Pro: you get to see how/where everything is set up

  Con: more typing

  added benefit: directly translates to a batch script

Hugging Face set up using a notebook

- **Many tutorials use Jupyter notebooks with “!pip install” commands**



```
In [ ]:  !pip install --upgrade 'huggingface_hub[pytorch,cli]'
         !pip install transformers
         !pip install accelerate
```

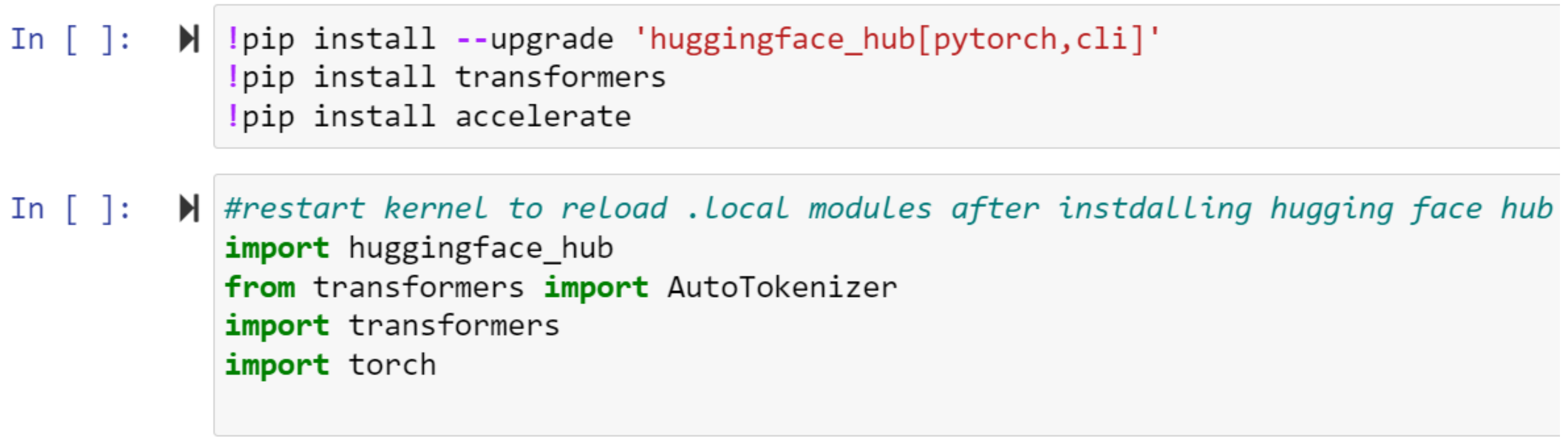
```
In [ ]:  #restart kernel to reload .local modules after instdalling hugging face hub
         import huggingface_hub
         from transformers import AutoTokenizer
         import transformers
         import torch
```

Packages will be in /home/userid/.local

After installing (~5 min) restart kernel to reset variables

Hugging Face set up using a notebook

- **Many tutorials use Jupyter notebooks with "!pip install" commands**



```
In [ ]:  !pip install --upgrade 'huggingface_hub[pytorch,cli]'
         !pip install transformers
         !pip install accelerate
```

```
In [ ]:  #restart kernel to reload .local modules after instdalling hugging face hub
         import huggingface_hub
         from transformers import AutoTokenizer
         import transformers
         import torch
```

Packages will be in /home/userid/.local

After installing, restart kernel to reset variables

BEWARE that "pip install" :

might create conflicts with other installs (try clear out/save previous '.local')

might try to uninstall a package on the system's Path path (try the –ignore-installed option)

might take a while

Otherwise it's easy

Hugging Face set up using the command line (instead of notebook)

Main Steps

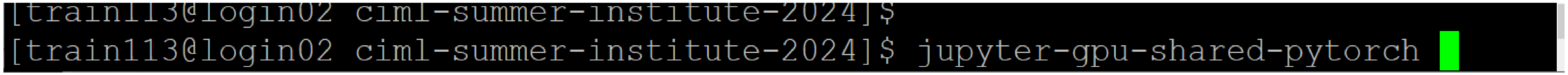
1. Request a GPU node
2. ssh into that node
3. Load modules
4. Run “singularity shell” command
5. Run “pip install” commands
6. Login to hugging face
7. Run your script

With practice you might prefer to work from command line to see all the details

Hugging Face set up using the command line

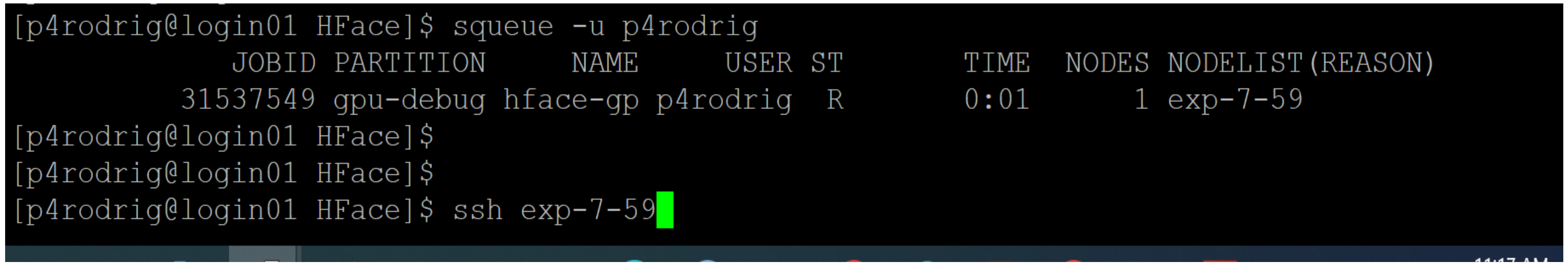
Main Steps

1. Request a GPU node



```
[train113@login02 ciml-summer-institute-2024]$
[train113@login02 ciml-summer-institute-2024]$ jupyter-gpu-shared-pytorch
```

2. ssh into that node

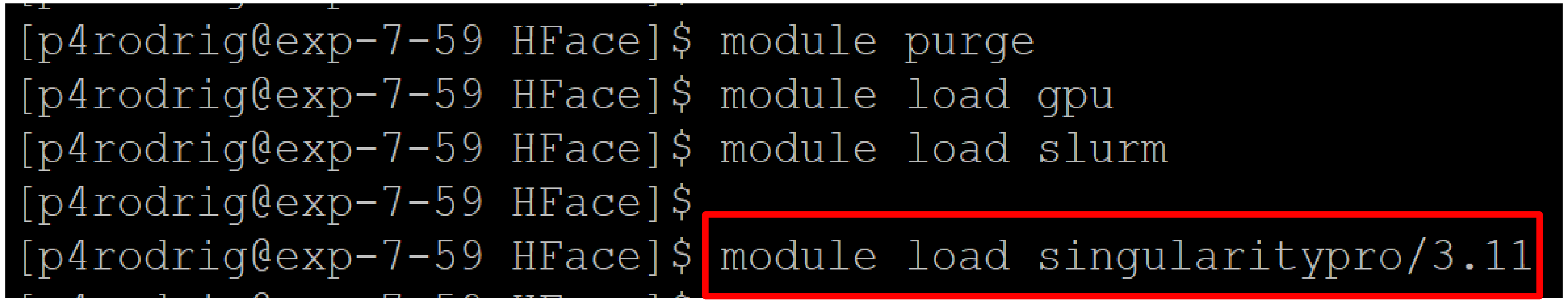


```
[p4rodrig@login01 HFace]$ squeue -u p4rodrig
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          31537549 gpu-debug hface-gp p4rodrig  R       0:01      1 exp-7-59
[p4rodrig@login01 HFace]$
[p4rodrig@login01 HFace]$
[p4rodrig@login01 HFace]$ ssh exp-7-59
```

Hugging Face set up using the command line

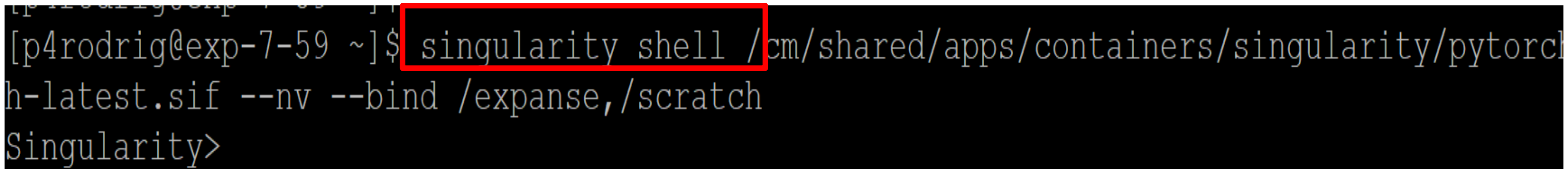
Main Steps

3. Load modules



```
[p4rodrig@exp-7-59 HFace]$ module purge
[p4rodrig@exp-7-59 HFace]$ module load gpu
[p4rodrig@exp-7-59 HFace]$ module load slurm
[p4rodrig@exp-7-59 HFace]$
[p4rodrig@exp-7-59 HFace]$ module load singularitypro/3.11
```

4. Run “singularity shell” command

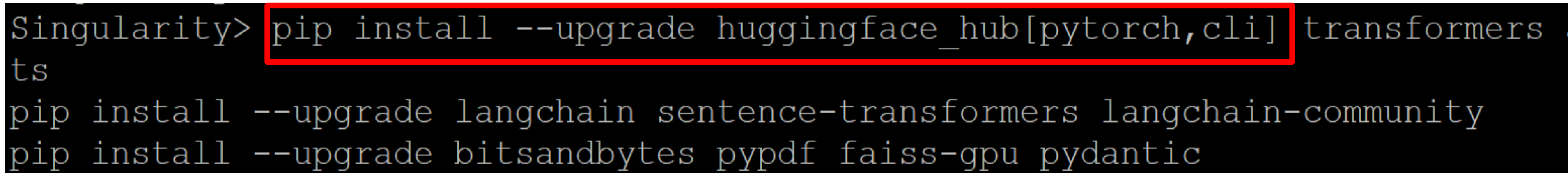


```
[p4rodrig@exp-7-59 ~]$ singularity shell /cm/shared/apps/containers/singularity/pytorc
h-latest.sif --nv --bind /expanse,/scratch
Singularity>
```

Hugging Face set up using the command line

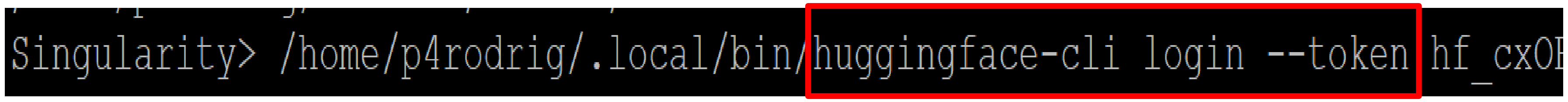
Main Steps

5. **Run "pip install" commands (takes ~5minutes)**



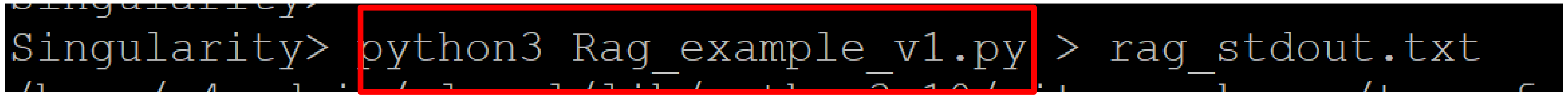
```
Singularity> pip install --upgrade huggingface_hub[pytorch,cli] transformers
ts
pip install --upgrade langchain sentence-transformers langchain-community
pip install --upgrade bitsandbytes pypdf faiss-gpu pydantic
```

6. **Login to hugging face**



```
Singularity> /home/p4rodrig/.local/bin/huggingface-cli login --token hf_cx0
```

7. **Run your script**

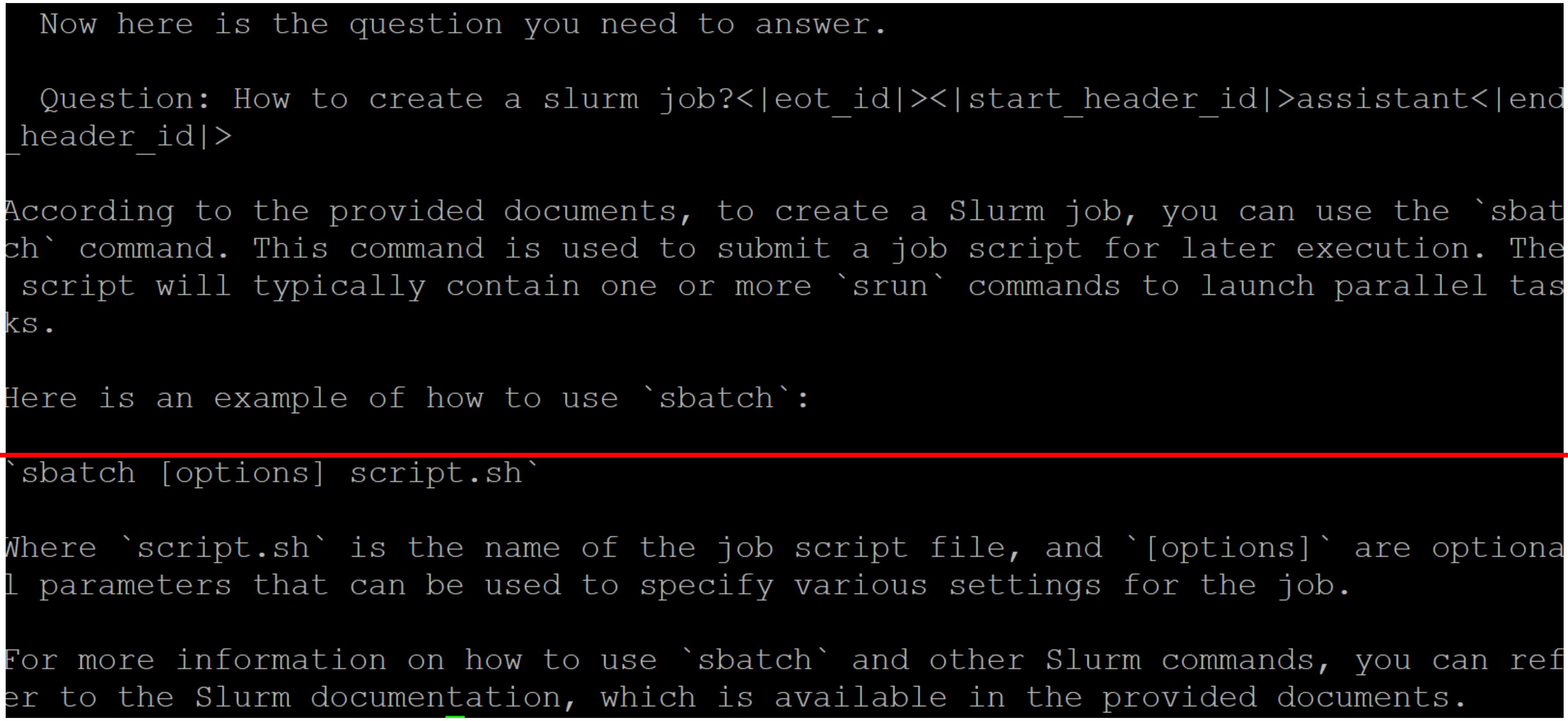


```
Singularity> python3 Rag_example_v1.py > rag_stdout.txt
```

Hugging Face set up using the command line

Other Steps

Review output:

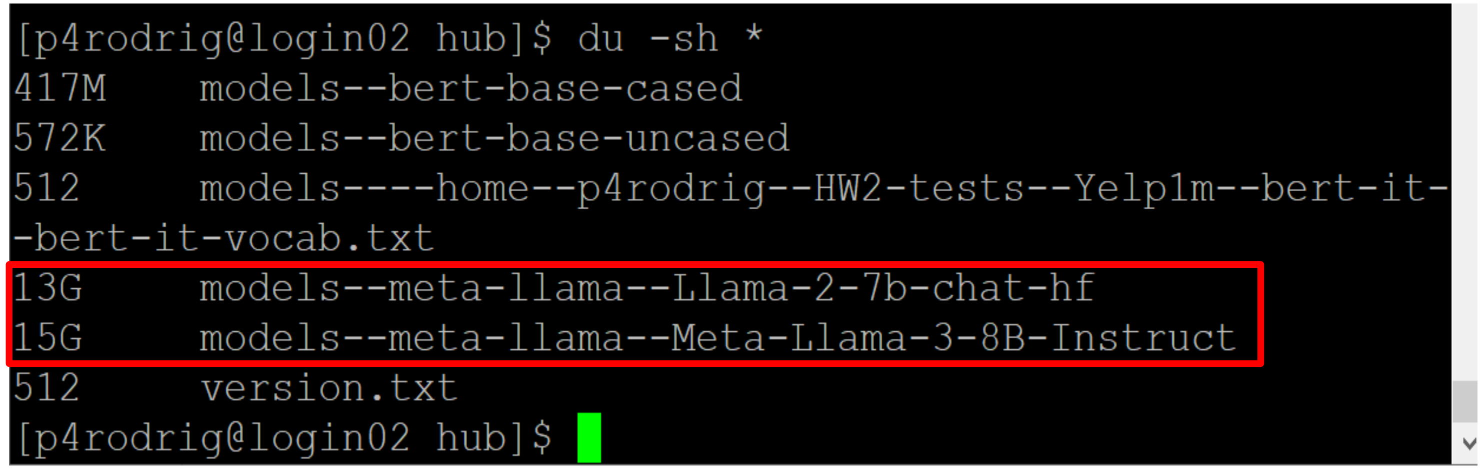


```
 Now here is the question you need to answer.

 Question: How to create a slurm job?<|eot_id|><|start_header_id|>assistant<|end
_header_id|>

According to the provided documents, to create a Slurm job, you can use the `sbat
ch` command. This command is used to submit a job script for later execution. The
 script will typically contain one or more `srun` commands to launch parallel tas
ks.

Here is an example of how to use `sbatch`:

`sbatch [options] script.sh`

Where `script.sh` is the name of the job script file, and `[options]` are optiona
l parameters that can be used to specify various settings for the job.

For more information on how to use `sbatch` and other Slurm commands, you can ref
er to the Slurm documentation, which is available in the provided documents.
```

Hugging Face set up using the command line

Other Steps

Review cached models in ~/.cach/huggingface/hub



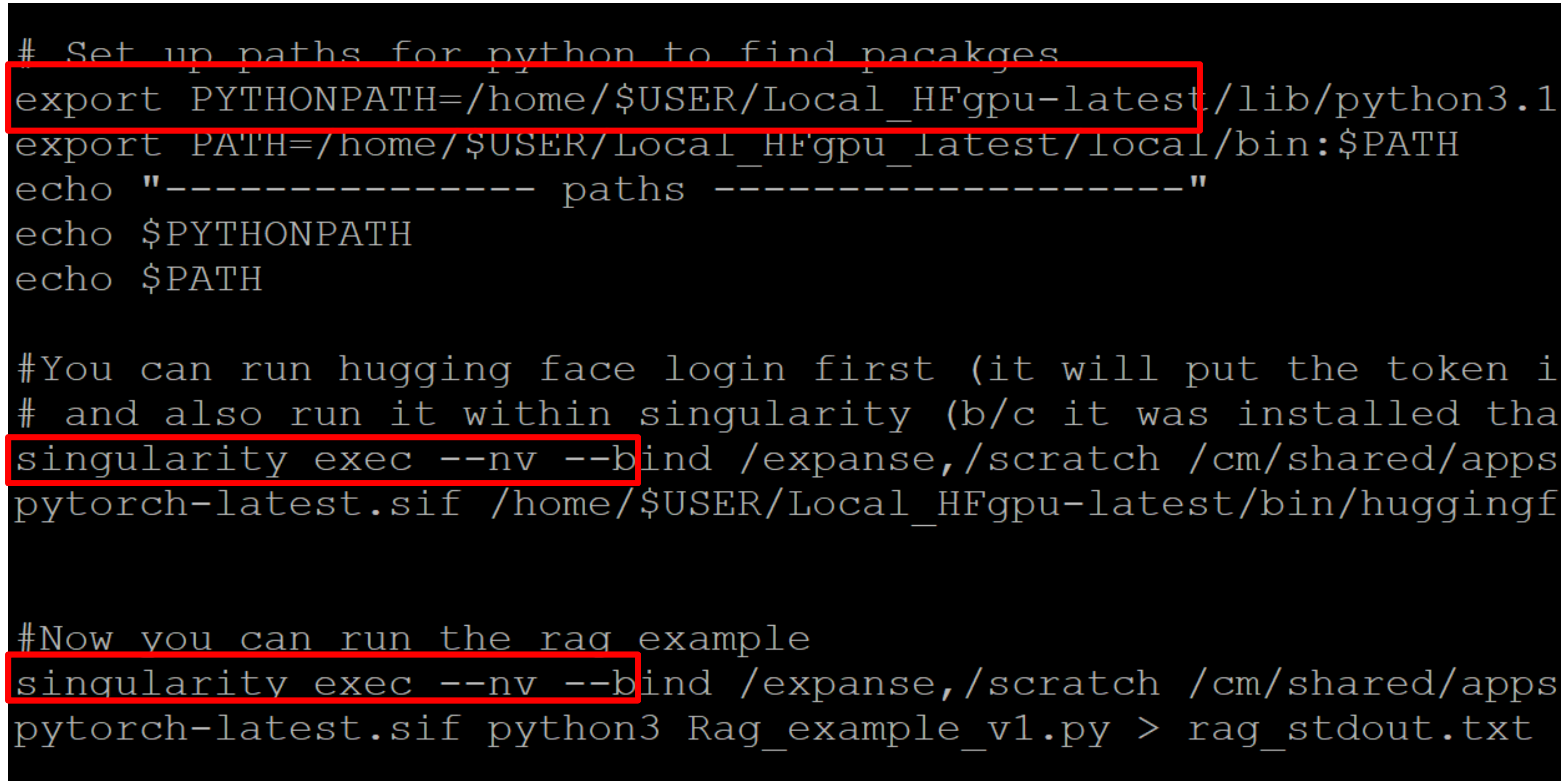
```
[p4rodrig@login02 hub]$ du -sh *
417M    models--bert-base-cased
572K    models--bert-base-uncased
512     models----home--p4rodrig--HW2-tests--Yelp1m--bert-it-
-bert-it-vocab.txt
13G     models--meta-llama--Llama-2-7b-chat-hf
15G     models--meta-llama--Meta-Llama-3-8B-Instruct
512     version.txt
[p4rodrig@login02 hub]$
```

Hugging Face in a batch script

Other Steps

move .local to new folder, and export PYTHONPATH to avoid potential conflicts

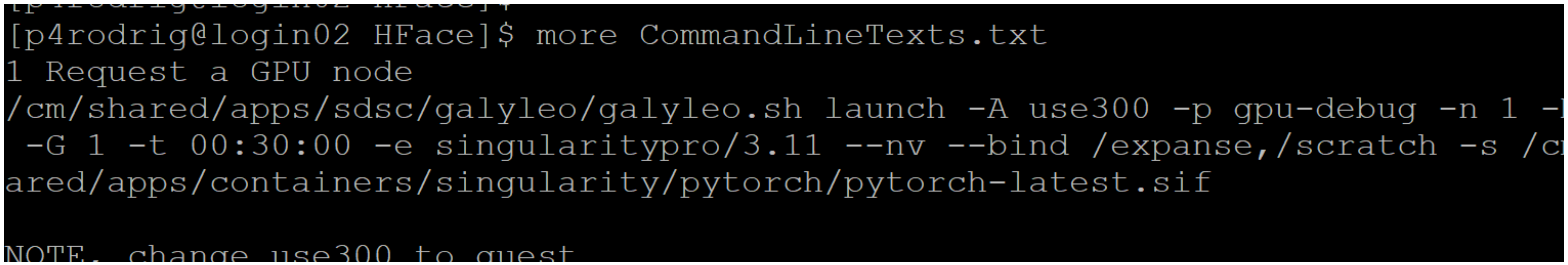
Use “singularity exec” to launch commands in a batch script



```
# Set up paths for python to find pacakges
export PYTHONPATH=/home/$USER/Local_HFgpu-latest/lib/python3.1
export PATH=/home/$USER/Local_HFgpu_latest/local/bin:$PATH
echo "---------------- paths --------------------"
echo $PYTHONPATH
echo $PATH

#You can run hugging face login first (it will put the token i
# and also run it within singularity (b/c it was installed tha
singularity exec --nv --bind /expanse,/scratch /cm/shared/apps
pytorch-latest.sif /home/$USER/Local_HFgpu-latest/bin/huggingf


#Now you can run the rag example
singularity exec --nv --bind /expanse,/scratch /cm/shared/apps
pytorch-latest.sif python3 Rag_example_v1.py > rag_stdout.txt
```

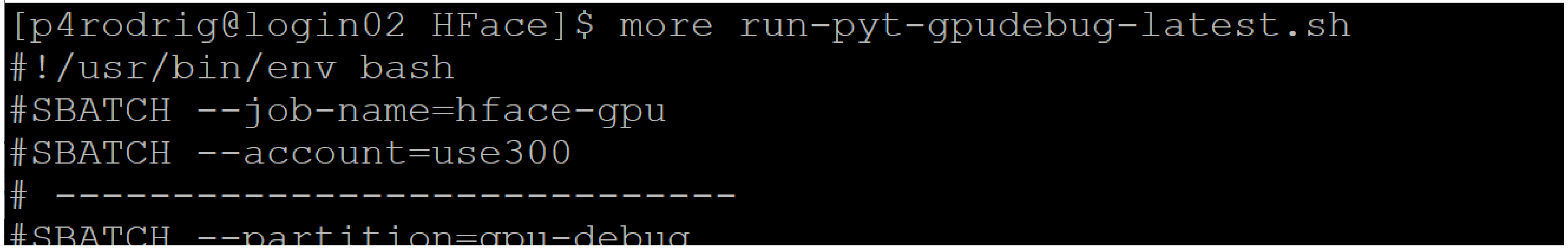
Text for command lines

- The above instructions are in “CommandLineTexts2.txt” file



```
[p4rodrig@login02 HFace]$ more CommandLineTexts.txt
1 Request a GPU node
/cm/shared/apps/sdsc/galyleo/galyleo.sh launch -A use300 -p gpu-debug -n 1 -
 -G 1 -t 00:30:00 -e singularitypro/3.11 --nv --bind /expanse,/scratch -s /c
ared/apps/containers/singularity/pytorch/pytorch-latest.sif

NOTE, change use300 to guest
```

- Also look at the run-pyt-gpu.. batch script

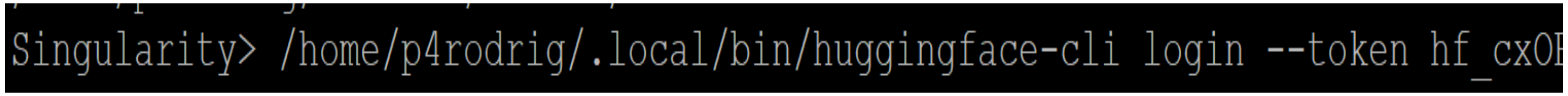


```
[p4rodrig@login02 HFace]$ more run-pyt-gpudebug-latest.sh
#!/usr/bin/env bash
#SBATCH --job-name=hface-gpu
#SBATCH --account=use300
# -----------------------------------
#SBATCH --partition=gpu-debug
```

Hugging Face set up using the command line

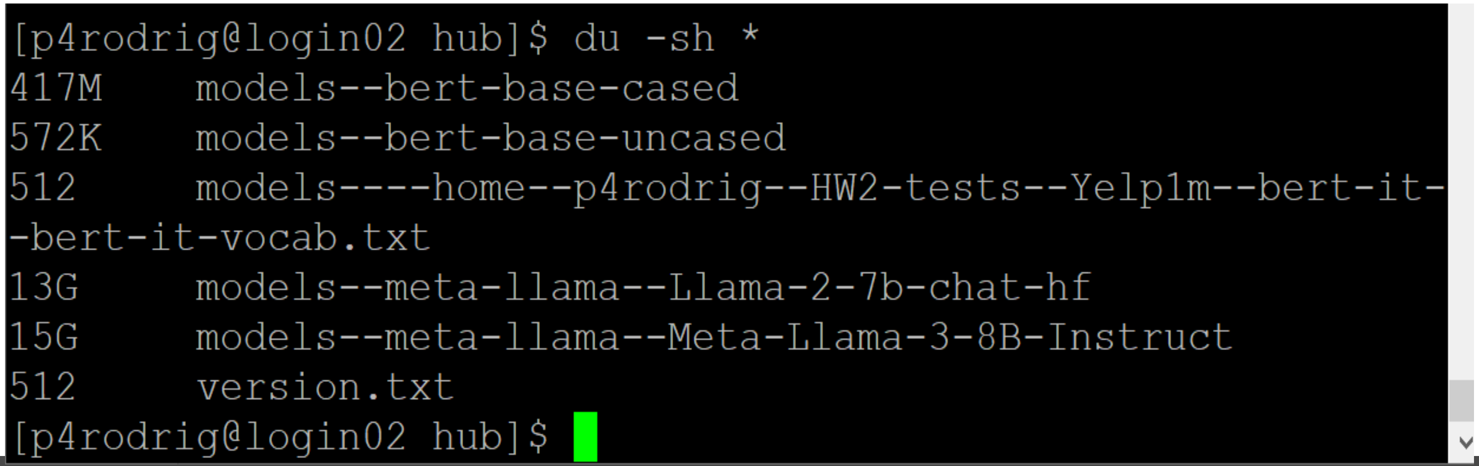
Other details like:

The command line command to login to hugging face



```
Singularity> /home/p4rodrig/.local/bin/huggingface-cli login --token hf_cxOI
```

Review cached models in ~/.cach/huggingface/hub



```
[p4rodrig@login02 hub]$ du -sh *
417M    models--bert-base-cased
572K    models--bert-base-uncased
512     models----home--p4rodrig--HW2-tests--Yelp1m--bert-it-
-bert-it-vocab.txt
13G     models--meta-llama--Llama-2-7b-chat-hf
15G     models--meta-llama--Meta-Llama-3-8B-Instruct
512     version.txt
[p4rodrig@login02 hub]$
```

end