AAMIR ABBAS

Advanced Big Data Analytics

UET Lahore

REG# 2023 MSDS 04

TASK 1

Q1.1 Respond with insights grounded in the terminology of Large Language Models.

(1). Can you provide a high-level overview of Transformers' architecture?

(2). What are the two approaches for evaluating language models in NLP, providing brief descriptions of each method along with highlighting their key distinctions?

(3). What is a token in the Large Language Models?

1). Overview of Transformers' Architecture:

Ans: Transformers are a type of neural network architecture that has proven to be highly effective in natural language processing (NLP) tasks and other machine learning applications. The architecture was introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017. Here's a high-level overview of the Transformer architecture:

1. **Self-Attention Mechanism:**

Allows the model to focus on different parts of the input sequence when making predictions.

2. **Multi-Head Attention:**

Uses multiple self-attention mechanisms to capture different aspects of the input.

3. **Positional Encoding:**

Adds information about the position of each word in the sequence.

4. **Feedforward Neural Network:**

Applies a simple neural network to process the attention outputs.

5. **Layer Normalization and Residual Connections:**

Helps in training deeper models by normalizing and connecting sub-layers.

6. **Encoder and Decoder Stacks:**

Multiple layers of the above components stacked together.

7. **Masked Self-Attention in Decoders:**

During training, prevents the model from looking ahead in the sequence.

8. **Output Layer:**

Produces the final predictions based on the processed information.

2). What are the two approaches for evaluating language models in NLP, providing brief descriptions of each method along with highlighting their key distinctions?

Ans: There are two ways to evaluate language models in NLP:

1. **Intrinsic Evaluation:**

What it does: Tests the model on specific language tasks like predicting the next word or recognizing parts of speech.

Why it's useful: Helps understand how well the model handles individual language aspects.

2. **Extrinsic Evaluation:**

What it does: Puts the model to the test in real-world applications, like translation or summarization.

Why it's useful: Shows how effective the model is in practical, broader language use.

Key Difference:

Intrinsic: Specific language tasks, direct metrics.

Extrinsic: Real-world applications, broader metrics.

Both evaluations together give a full picture of how good a language model is at understanding and using language.

---

3). What is a token in the Large Language Models?

Ans: Tokens are the fundamental building blocks that LLMs use to process and understand language. Like words in human language, they are the smallest meaningful units that LLMs can work with.

Key points:

- Not always individual words: Tokens can be words, but they can also be subwords, characters, or even entire phrases, depending on the model and task.
- Created through tokenization: The process of dividing text into tokens is called tokenization.
- Vital for language understanding: LLMs use tokens to represent and analyze the structure and meaning of language.
- Input and output: Tokens serve as both the input and output for LLMs during tasks like text generation and translation.

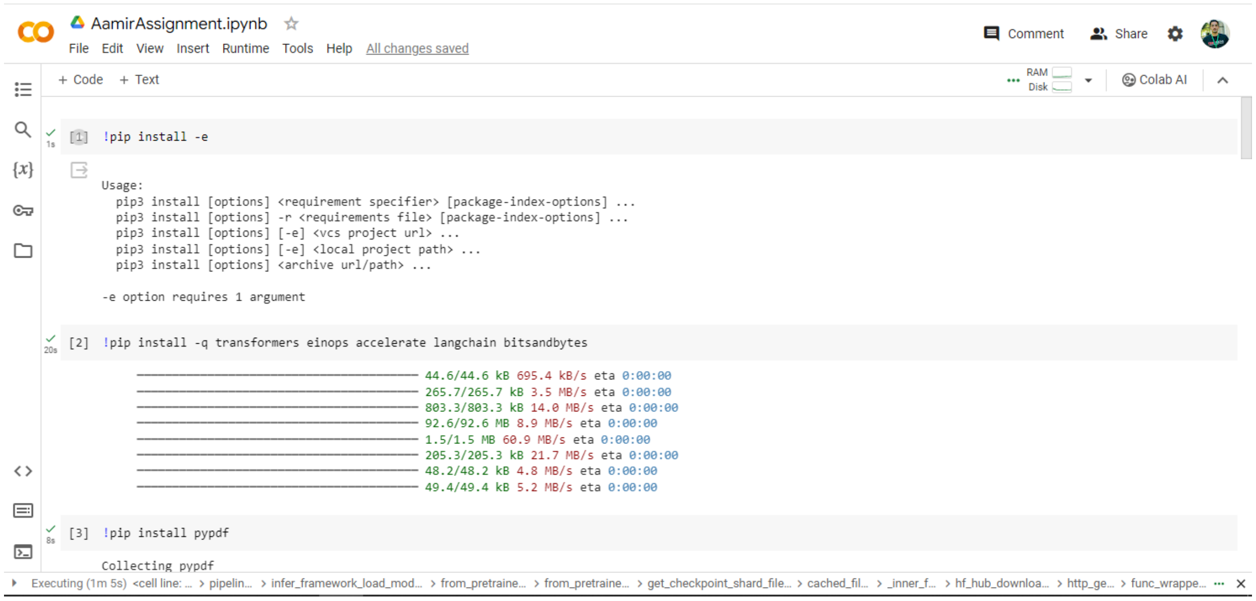
Example:

Consider the sentence "I love reading books." Here are possible tokenizations:

Word-based: ["I", "love", "reading", "books"]

Q1.2 Read through the tutorial slides and deploy Llama 2 on Google Colab and get inference from the model. (10 pts)

(1). Provide screenshots of the results after you successfully download the model and see the text generated.

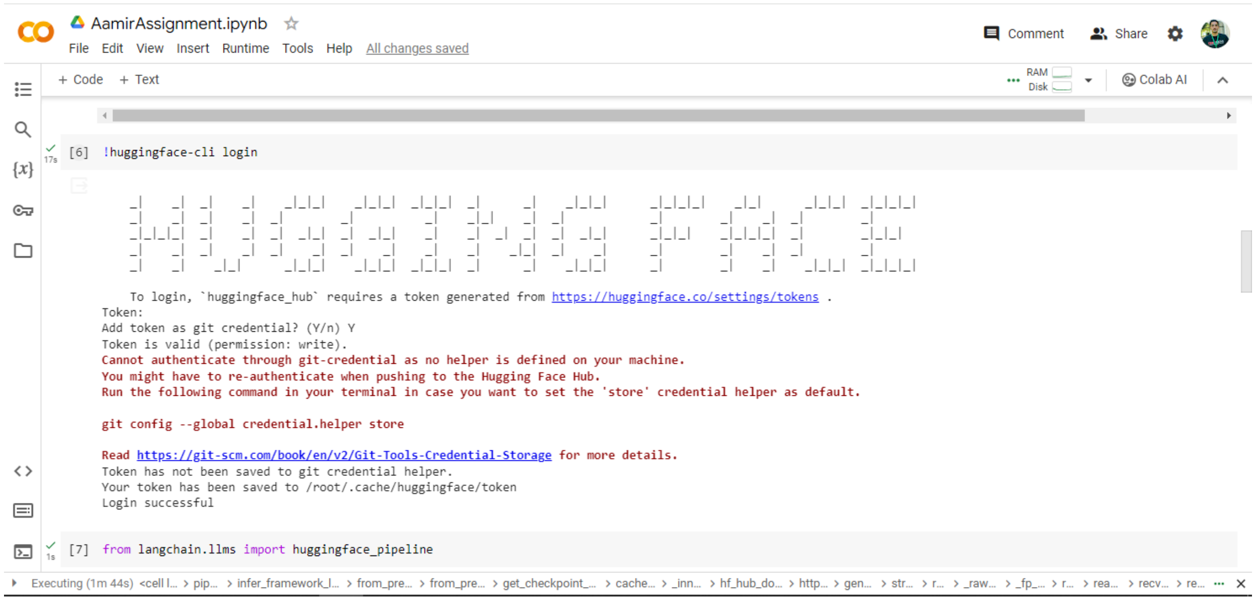


```
[1] !pip install -e

Usage:
  pip3 install [options] <requirement specifier> [package-index-options] ...
  pip3 install [options] -r <requirements file> [package-index-options] ...
  pip3 install [options] [-e] <vcs project url> ...
  pip3 install [options] [-e] <local project path> ...
  pip3 install [options] <archive url/path> ...

-e option requires 1 argument
```

```
[2] !pip install -q transformers einops accelerate langchain bitsandbytes

     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.6/44.6 kB 695.4 kB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 265.7/265.7 kB 3.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 803.3/803.3 kB 14.0 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 92.6/92.6 MB 8.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.5/1.5 MB 60.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 205.3/205.3 kB 21.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 48.2/48.2 kB 4.8 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.4/49.4 kB 5.2 MB/s eta 0:00:00
```

```
[3] !pip install pypdf

Collecting pypdf
```

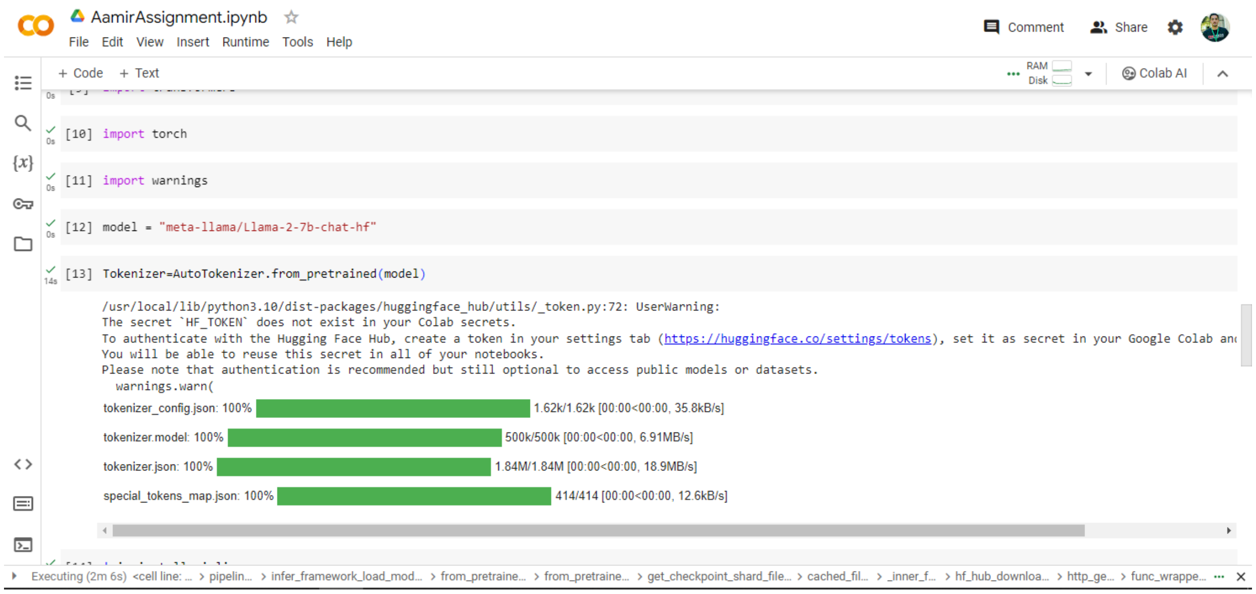
```
Executing (1m 5s) <cell line: ... > pipelin... > infer_framework_load_mod... > from_pretraine... > from_pretraine... > get_checkpoint_shard_file... > cached_fil... > _inner_f... > hf_hub_downloa... > http_ge... > func_wrappe...
```



```
[6] !huggingface-cli login

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

    To login, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tokens .
Token:
Add token as git credential? (Y/n) Y
Token is valid (permission: write).
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
```

```
[7] from langchain.llms import huggingface_pipeline
```

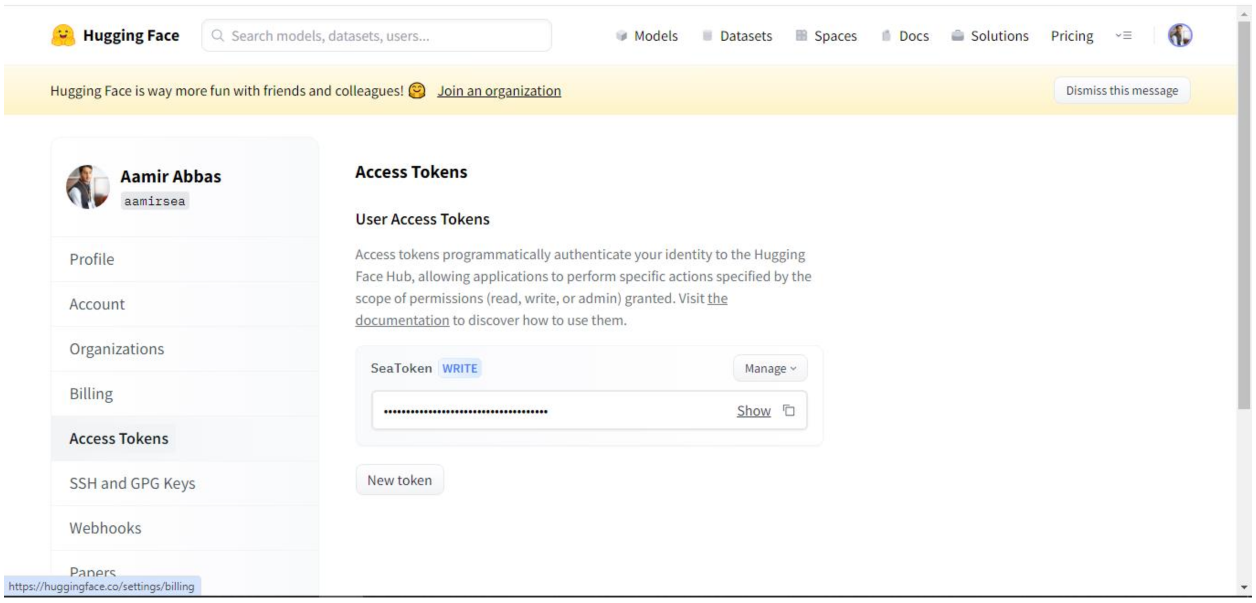
```
Executing (1m 44s) <cell l... > pip... > infer_framework_l... > from_pre... > from_pre... > get_checkpoint_... > cache... > _inn... > hf_hub_do... > http... > gen... > str... > r... > _raw... > _fp_... > r... > rea... > recv... > re...
```



```
[10] import torch

[11] import warnings

[12] model = "meta-llama/Llama-2-7b-chat-hf"

[13] Tokenizer=AutoTokenizer.from_pretrained(model)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:72: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab an
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 100% 1.62k/1.62k [00:00<00:00, 35.8kB/s]
tokenizer.model: 100% 500k/500k [00:00<00:00, 6.91MB/s]
tokenizer.json: 100% 1.84M/1.84M [00:00<00:00, 18.9MB/s]
special_tokens_map.json: 100% 414/414 [00:00<00:00, 12.6kB/s]
```

```
Executing (2m 6s) <cell line: ... > pipelin... > infer_framework_load_mod... > from_pretraine... > from_pretraine... > get_checkpoint_shard_file... > cached_fil... > _inner_f... > hf_hub_downloa... > http_ge... > func_wrappe...
```



Hugging Face is way more fun with friends and colleagues! Join an organization

Dismiss this message

Aamir Abbas
aamirsea

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

Access Tokens

User Access Tokens

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

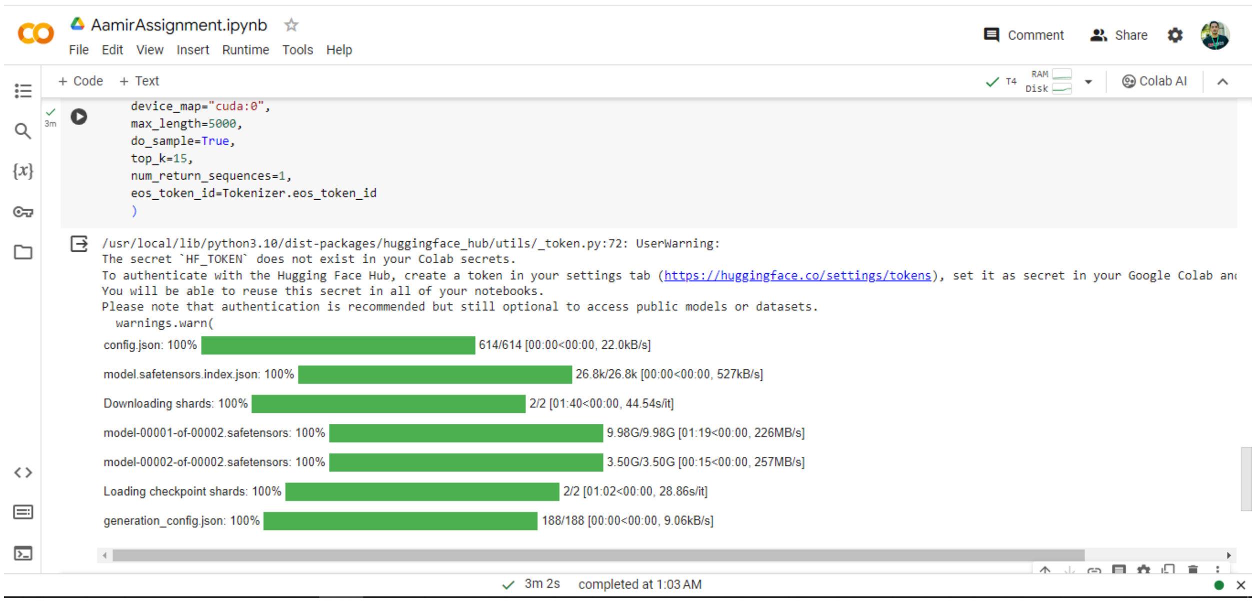
SeaToken WRITE Manage

•••••••••••••••••••••••••••••••••••••• Show

New token

https://huggingface.co/settings/billing

(2). Change the "max_length" variable in pipline and observe the difference.



```
    device_map="cuda:0",
    max_length=5000,
    do_sample=True,
    top_k=15,
    num_return_sequences=1,
    eos_token_id=Tokenizer.eos_token_id
    )
```

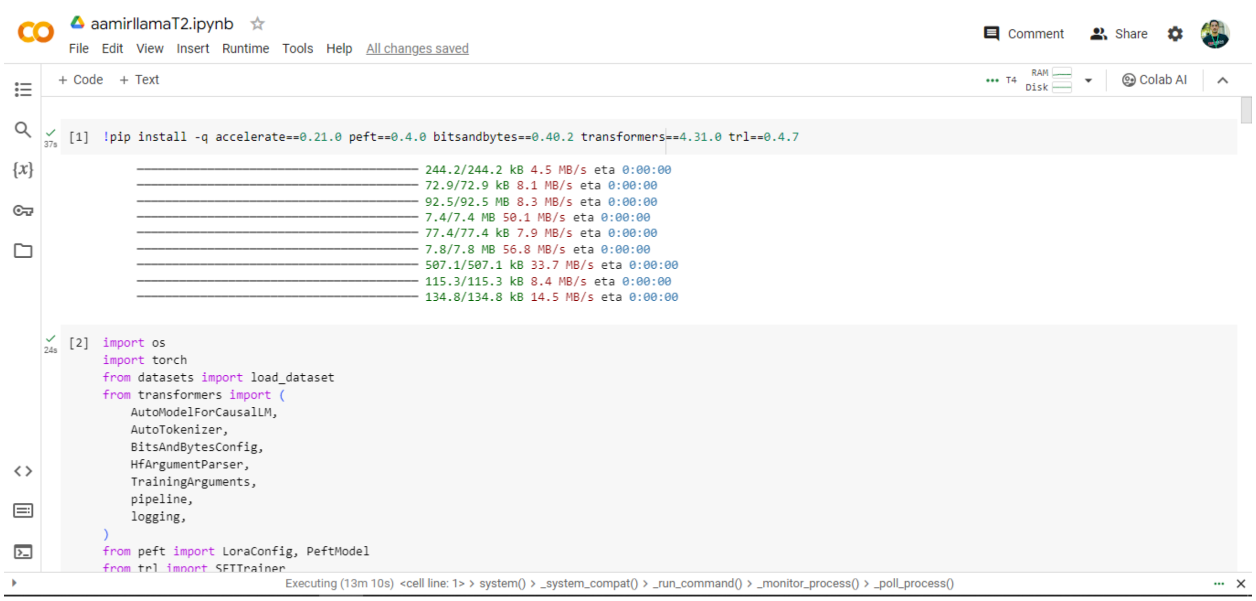
```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:72: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab an
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100% 614/614 [00:00<00:00, 22.0kB/s]
model.safetensors.index.json: 100% 26.8k/26.8k [00:00<00:00, 527kB/s]
Downloading shards: 100% 2/2 [01:40<00:00, 44.54s/it]
model-00001-of-00002.safetensors: 100% 9.98G/9.98G [01:19<00:00, 226MB/s]
model-00002-of-00002.safetensors: 100% 3.50G/3.50G [00:15<00:00, 257MB/s]
Loading checkpoint shards: 100% 2/2 [01:02<00:00, 28.86s/it]
generation_config.json: 100% 188/188 [00:00<00:00, 9.06kB/s]
```

✓ 3m 2s completed at 1:03 AM

Task 2

In this part you are going to fine tuning Llama 2 models based on OpenAssistant dataset.

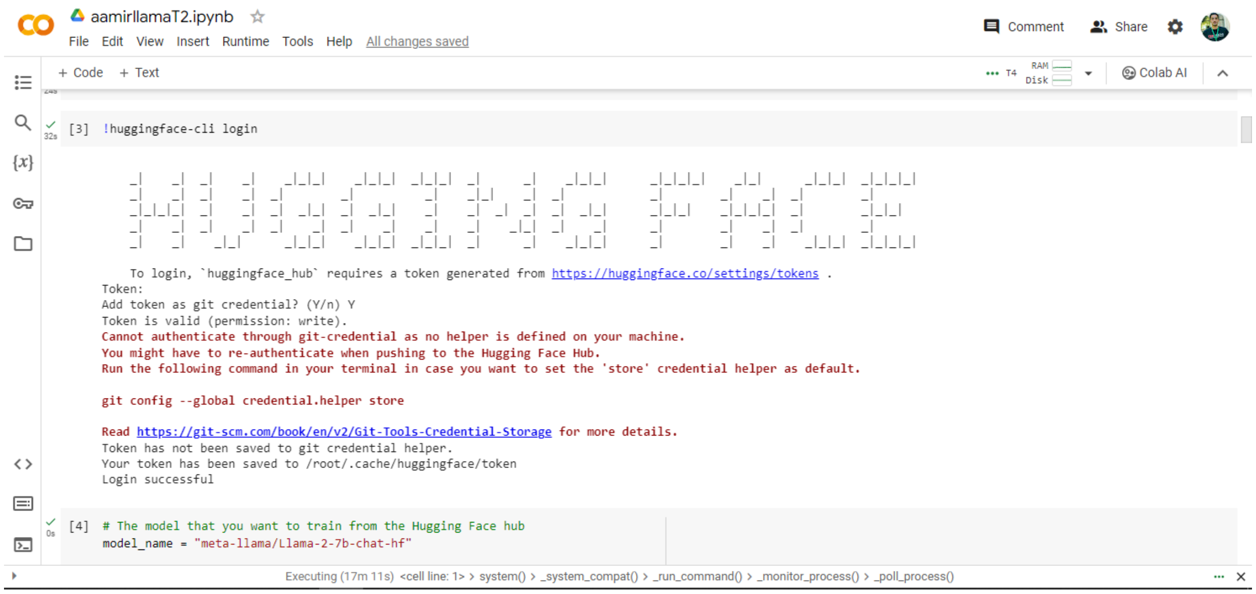
Q2.1 Write comments for each line of code and succintly explain what it is doing.



```
[1] !pip install -q accelerate==0.21.0 peft==0.4.0 bitsandbytes==0.40.2 transformers==4.31.0 trl==0.4.7

    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 244.2/244.2 kB 4.5 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 72.9/72.9 kB 8.1 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 92.5/92.5 MB 8.3 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.4/7.4 MB 50.1 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 77.4/77.4 kB 7.9 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.8/7.8 MB 56.8 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 507.1/507.1 kB 33.7 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 115.3/115.3 kB 8.4 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 14.5 MB/s eta 0:00:00
```

```
[2] import os
    import torch
    from datasets import load_dataset
    from transformers import (
        AutoModelForCausalLM,
        AutoTokenizer,
        BitsAndBytesConfig,
        HfArgumentParser,
        TrainingArguments,
        pipeline,
        logging,
    )
    from peft import LoraConfig, PeftModel
    from trl import SFTTrainer
```

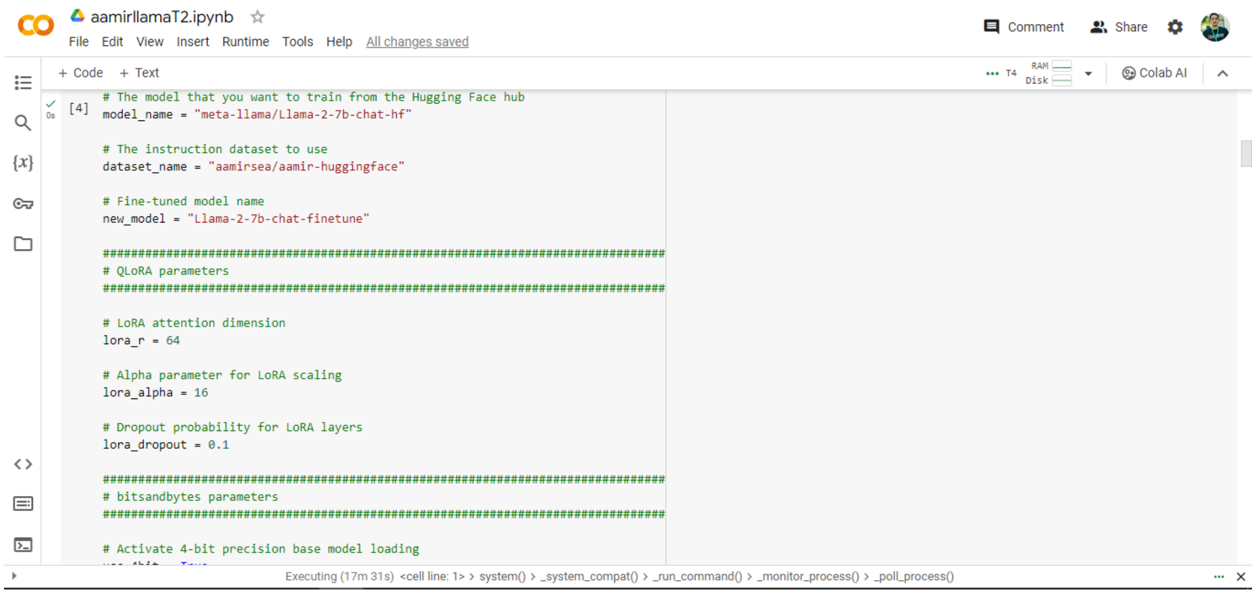
Executing (13m 10s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()



```
[3] !huggingface-cli login

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

    To login, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tokens .
Token:
Add token as git credential? (Y/n) Y
Token is valid (permission: write).
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
```

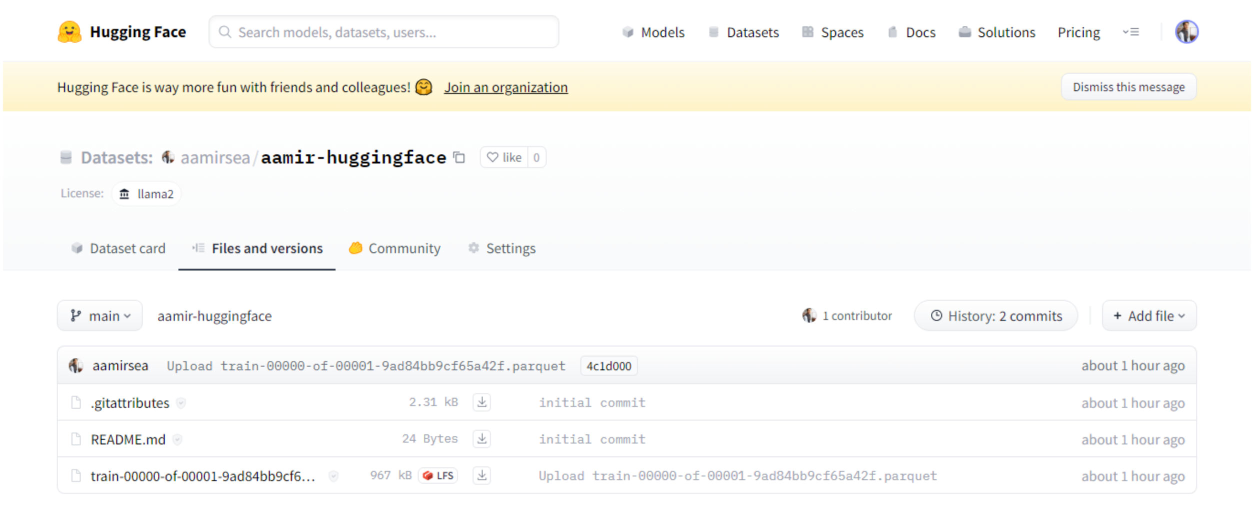
```
[4] # The model that you want to train from the Hugging Face hub
    model_name = "meta-llama/Llama-2-7b-chat-hf"
```

Executing (17m 11s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()



```
[4] # The model that you want to train from the Hugging Face hub
    model_name = "meta-llama/Llama-2-7b-chat-hf"

    # The instruction dataset to use
    dataset_name = "aamirsea/aamir-huggingface"

    # Fine-tuned model name
    new_model = "Llama-2-7b-chat-finetune"

    ################################################################################
    # QLoRA parameters
    ################################################################################

    # LoRA attention dimension
    lora_r = 64

    # Alpha parameter for LoRA scaling
    lora_alpha = 16

    # Dropout probability for LoRA layers
    lora_dropout = 0.1

    ################################################################################
    # bitsandbytes parameters
    ################################################################################

    # Activate 4-bit precision base model loading
```

Executing (17m 31s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()



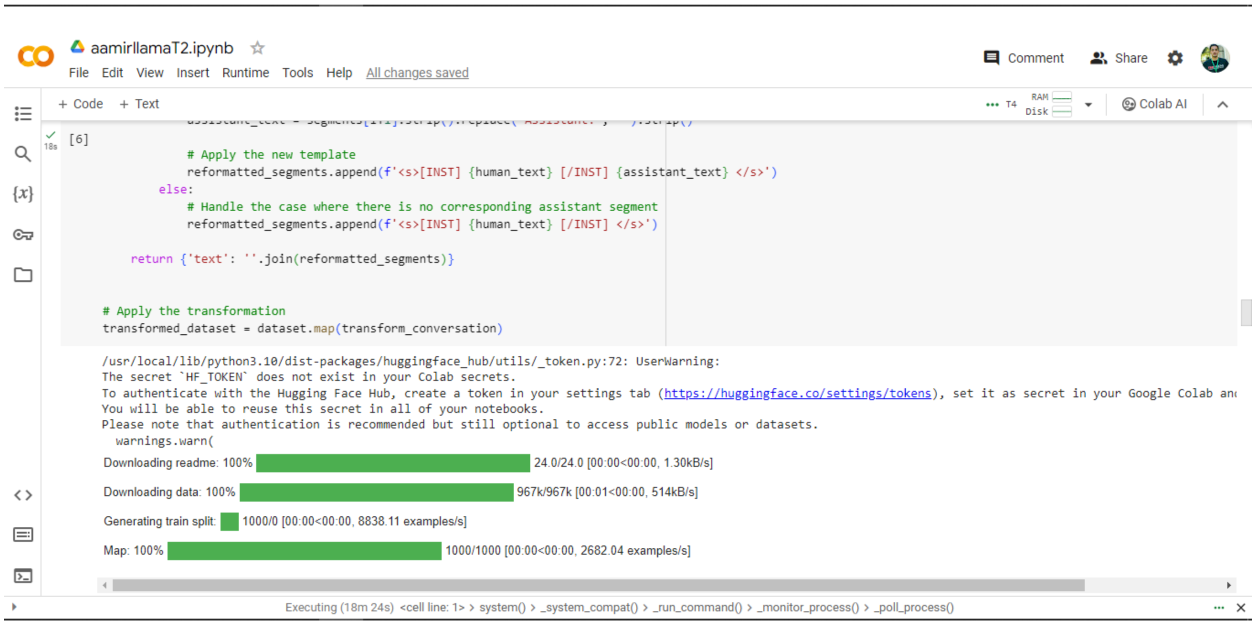
Hugging Face is way more fun with friends and colleagues! Join an organization

Datasets: aamirsea / aamir-huggingface

License: llama2

aamir-huggingface

aamirsea Upload train-00000-of-00001-9ad84bb9cf65a42f.parquet 4c1d000			about 1 hour ago
.gitattributes	2.31 kB	initial commit	about 1 hour ago
README.md	24 Bytes	initial commit	about 1 hour ago
train-00000-of-00001-9ad84bb9cf6…	967 kB LFS	Upload train-00000-of-00001-9ad84bb9cf65a42f.parquet	about 1 hour ago

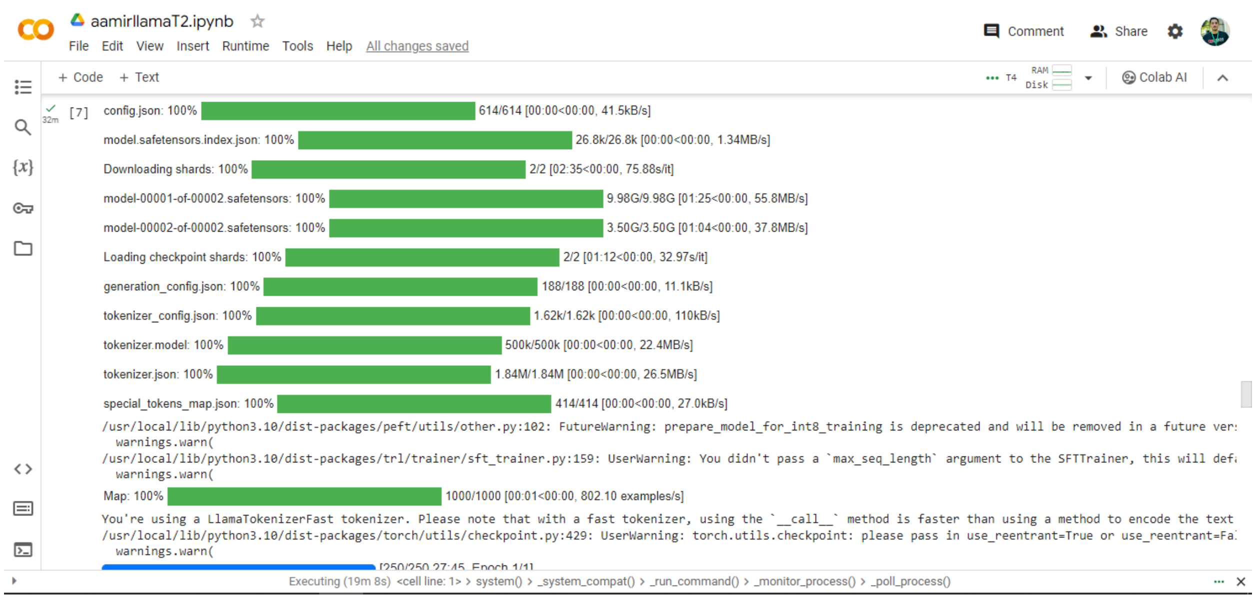


```
[6]
            # Apply the new template
            reformatted_segments.append(f'<s>[INST] {human_text} [/INST] {assistant_text} </s>')
        else:
            # Handle the case where there is no corresponding assistant segment
            reformatted_segments.append(f'<s>[INST] {human_text} [/INST] </s>')

    return {'text': ''.join(reformatted_segments)}


# Apply the transformation
transformed_dataset = dataset.map(transform_conversation)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:72: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab an
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
Downloading readme: 100% 24.0/24.0 [00:00<00:00, 1.30kB/s]
Downloading data: 100% 967k/967k [00:01<00:00, 514kB/s]
Generating train split: 1000/0 [00:00<00:00, 8838.11 examples/s]
Map: 100% 1000/1000 [00:00<00:00, 2682.04 examples/s]
```

Executing (18m 24s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()

Q2.2 Make a training loss plot.



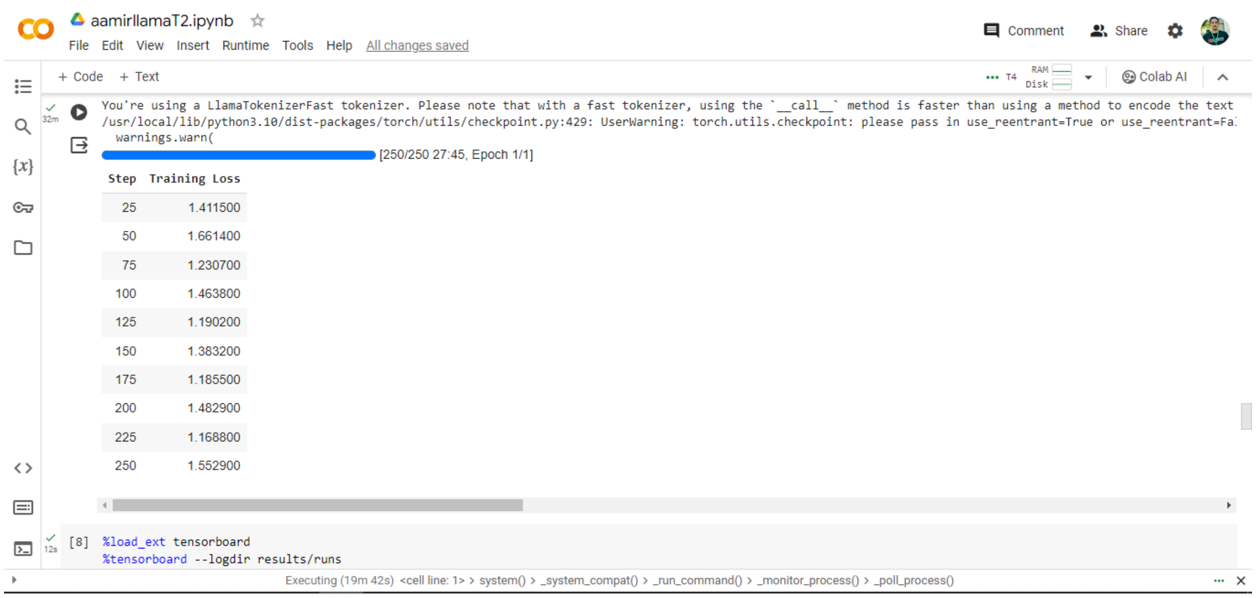
aamirllamaT2.ipynb

File Edit View Insert Runtime Tools Help All changes saved

[7]

```
config.json: 100%  614/614 [00:00<00:00, 41.5kB/s]
model.safetensors.index.json: 100%  26.8k/26.8k [00:00<00:00, 1.34MB/s]
Downloading shards: 100%  2/2 [02:35<00:00, 75.88s/it]
model-00001-of-00002.safetensors: 100%  9.98G/9.98G [01:25<00:00, 55.8MB/s]
model-00002-of-00002.safetensors: 100%  3.50G/3.50G [01:04<00:00, 37.8MB/s]
Loading checkpoint shards: 100%  2/2 [01:12<00:00, 32.97s/it]
generation_config.json: 100%  188/188 [00:00<00:00, 11.1kB/s]
tokenizer_config.json: 100%  1.62k/1.62k [00:00<00:00, 110kB/s]
tokenizer.model: 100%  500k/500k [00:00<00:00, 22.4MB/s]
tokenizer.json: 100%  1.84M/1.84M [00:00<00:00, 26.5MB/s]
special_tokens_map.json: 100%  414/414 [00:00<00:00, 27.0kB/s]
/usr/local/lib/python3.10/dist-packages/peft/utils/other.py:102: FutureWarning: prepare_model_for_int8_training is deprecated and will be removed in a future ver
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/trl/trainer/sft_trainer.py:159: UserWarning: You didn't pass a `max_seq_length` argument to the SFTTrainer, this will def
  warnings.warn(
Map: 100%  1000/1000 [00:01<00:00, 802.10 examples/s]
You're using a LlamaTokenizerFast tokenizer. Please note that with a fast tokenizer, using the `__call__` method is faster than using a method to encode the text
/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=Fa
  warnings.warn(
[250/250 27:45, Epoch 1/1]
```

Executing (19m 8s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()



aamirllamaT2.ipynb

File Edit View Insert Runtime Tools Help All changes saved

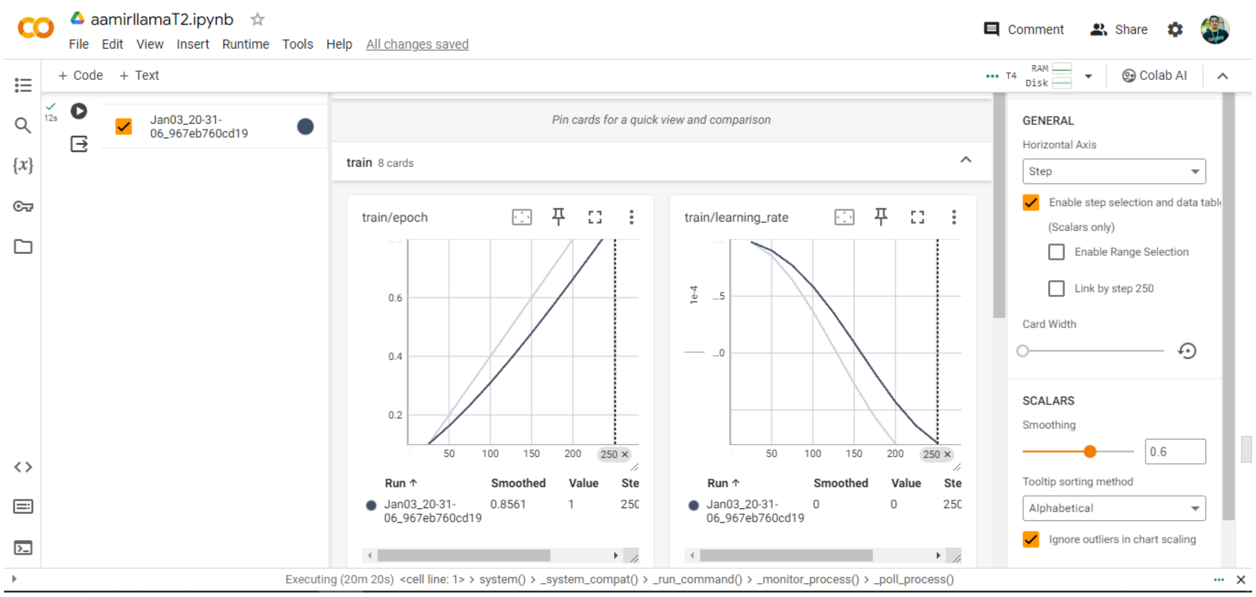
```
You're using a LlamaTokenizerFast tokenizer. Please note that with a fast tokenizer, using the `__call__` method is faster than using a method to encode the text
/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=Fa
  warnings.warn(
[250/250 27:45, Epoch 1/1]
```

Step	Training Loss
25	1.411500
50	1.661400
75	1.230700
100	1.463800
125	1.190200
150	1.383200
175	1.185500
200	1.482900
225	1.168800
250	1.552900

[8]

```
%load_ext tensorboard
%tensorboard --logdir results/runs
```

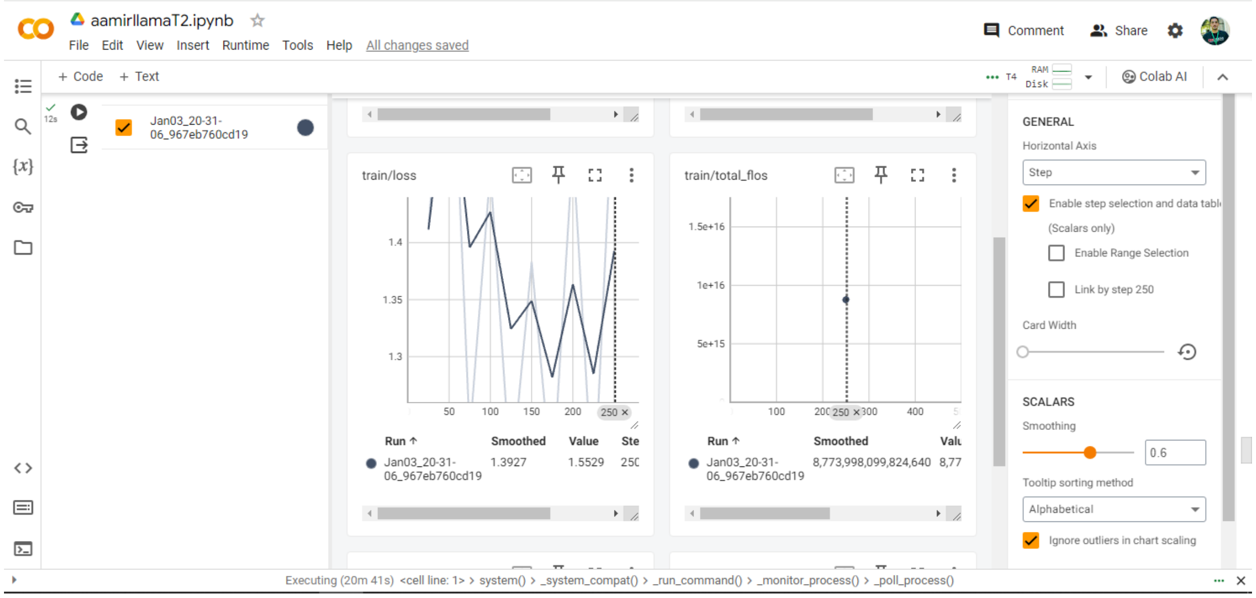
Executing (19m 42s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()



Run ↑	Smoothed	Value	Step
Jan03_20-31-06_967eb760cd19	0.8561	1	250

Run ↑	Smoothed	Value	Step
Jan03_20-31-06_967eb760cd19	0	0	250

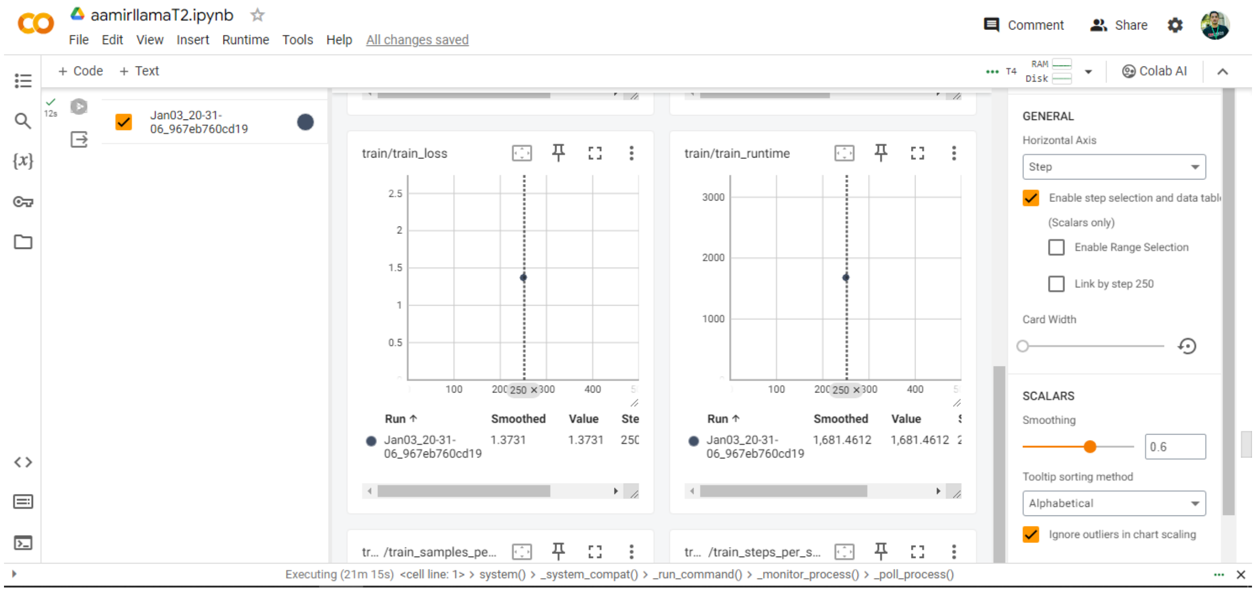
```
Executing (20m 20s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()
```



Run ↑	Smoothed	Value	Step
Jan03_20-31-06_967eb760cd19	1.3927	1.5529	250

Run ↑	Smoothed	Value
Jan03_20-31-06_967eb760cd19	8,773,998,099,824,640	8,77

```
Executing (20m 41s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()
```


aamirllamaT2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
T4 RAM Disk
Colab AI
Jan03_20-31-06_967eb760cd19
GENERAL
Horizontal Axis
Step
Enable step selection and data table (Scalars only)
Enable Range Selection
Link by step 250
Card Width
SCALARS
Smoothing
0.6
Tooltip sorting method
Alphabetical
Ignore outliers in chart scaling
tr... /train_samples_pe...
tr... /train_steps_per_s...

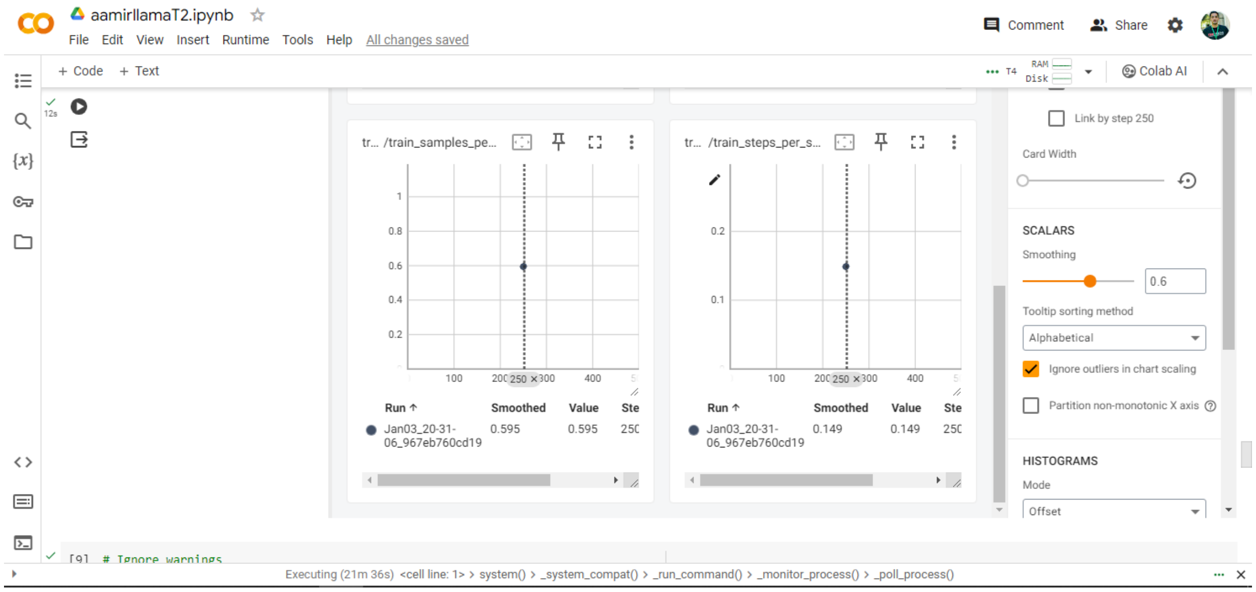
train/train_loss

Run ↑	Smoothed	Value	Ste
● Jan03_20-31-06_967eb760cd19	1.3731	1.3731	250

train/train_runtime

Run ↑	Smoothed	Value	S
● Jan03_20-31-06_967eb760cd19	1,681.4612	1,681.4612	2

```
Executing (21m 15s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()
```



tr... /train_samples_pe...

Run ↑	Smoothed	Value	Ste
● Jan03_20-31-06_967eb760cd19	0.595	0.595	250

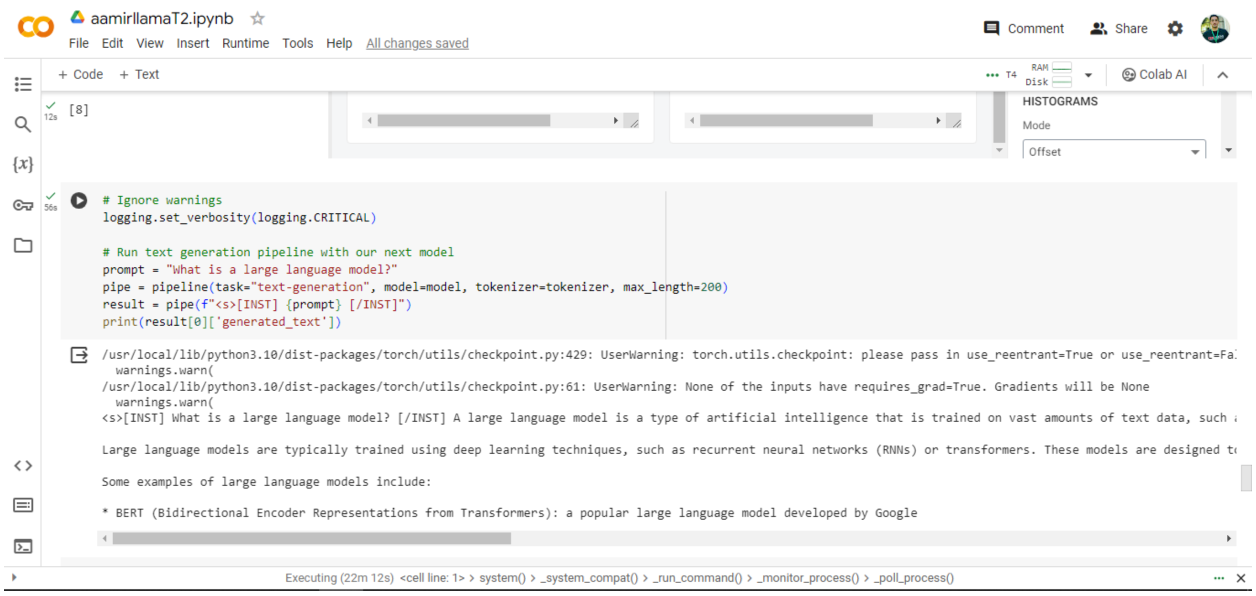
tr... /train_steps_per_s...

Run ↑	Smoothed	Value	Ste
● Jan03_20-31-06_967eb760cd19	0.149	0.149	250

```
[9]  # Ignore warnings
```

```
Executing (21m 36s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()
```

Q2.3 Use the text generation pipeline to ask questions like “What is a large language model?”



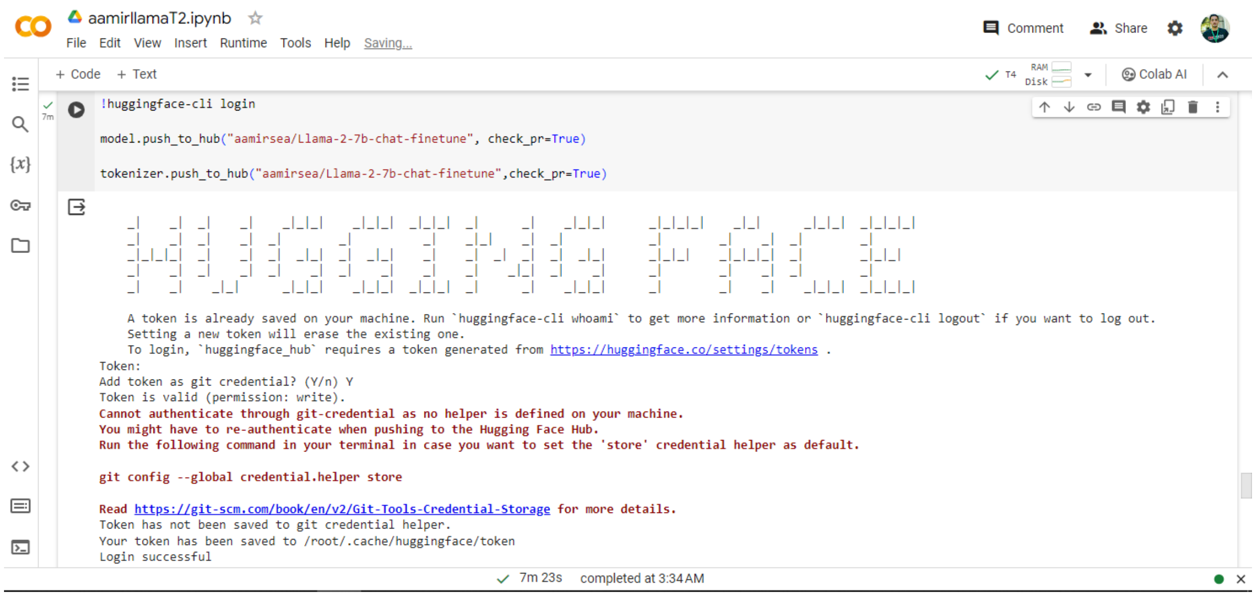
```
[8]
```

```python
# Ignore warnings
logging.set_verbosity(logging.CRITICAL)

# Run text generation pipeline with our next model
prompt = "What is a large language model?"
pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=200)
result = pipe(f"<s>[INST] {prompt} [/INST]")
print(result[0]['generated_text'])
```

```
/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=Fa
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:61: UserWarning: None of the inputs have requires_grad=True. Gradients will be None
  warnings.warn(
<s>[INST] What is a large language model? [/INST] A large language model is a type of artificial intelligence that is trained on vast amounts of text data, such

Large language models are typically trained using deep learning techniques, such as recurrent neural networks (RNNs) or transformers. These models are designed t

Some examples of large language models include:

* BERT (Bidirectional Encoder Representations from Transformers): a popular large language model developed by Google
```

```
Executing (22m 12s) <cell line: 1> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()
```

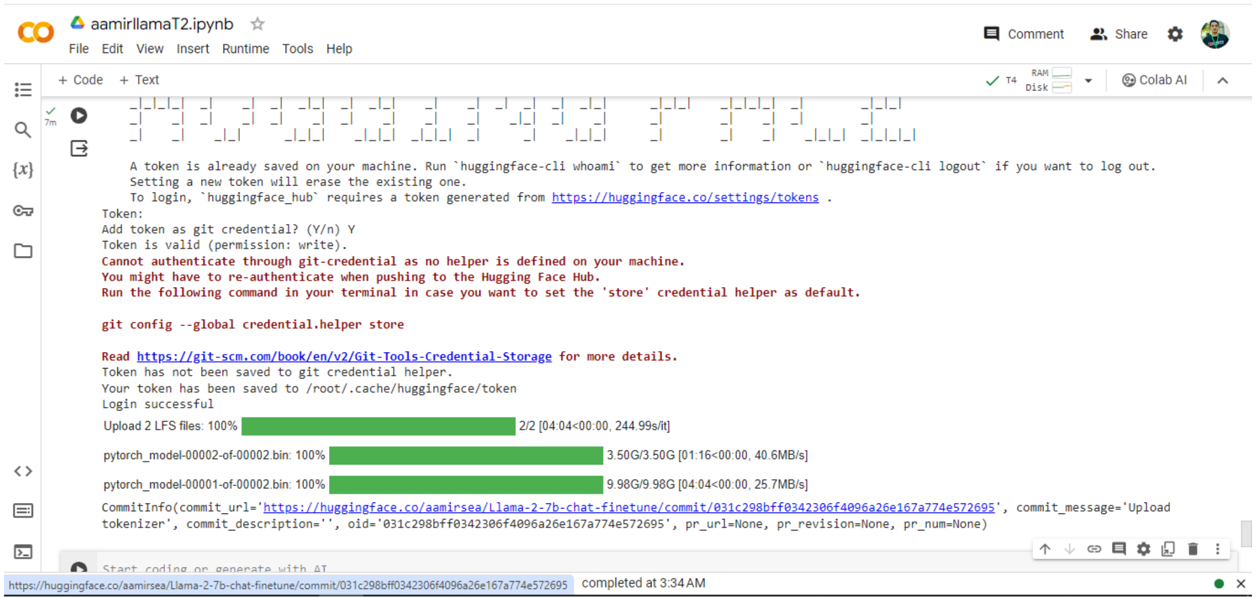
Q 2.4 Store fine-tuning Llama2 Model and push Model to your Hugging Face Hub. Provide screenshot of your Hugging Face model page.



```python
!huggingface-cli login

model.push_to_hub("aamirsea/Llama-2-7b-chat-finetune", check_pr=True)

tokenizer.push_to_hub("aamirsea/Llama-2-7b-chat-finetune",check_pr=True)
```

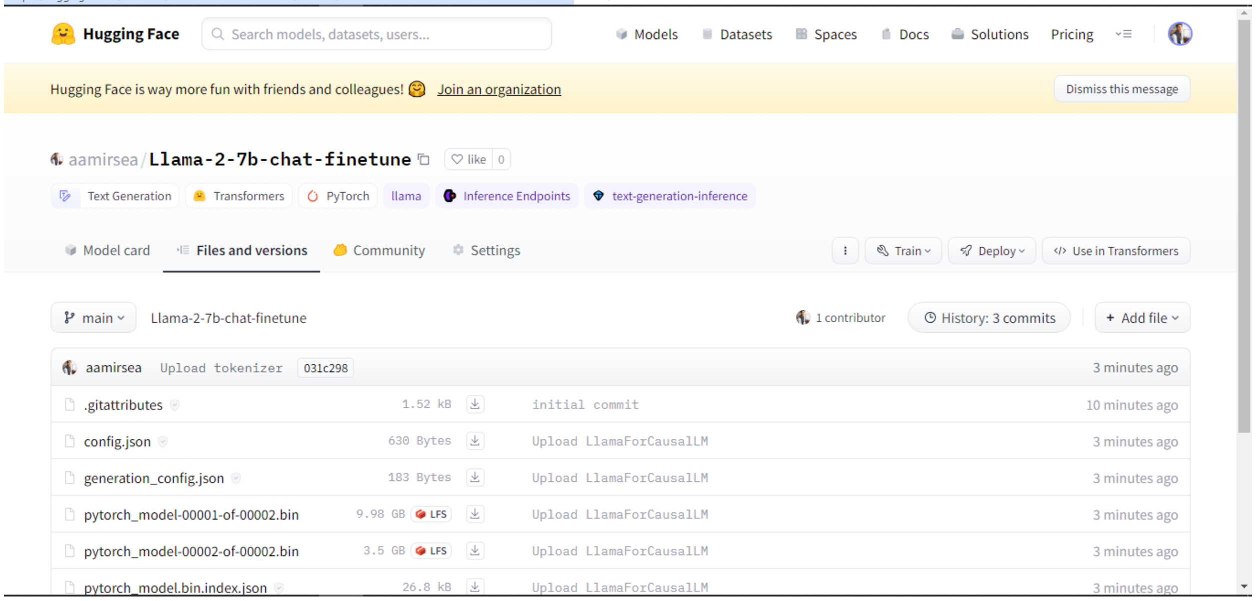
```
    _|    _|  _|    _|    _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|_|_|_|    _|_|      _|_|_|  _|_|_|_|
    _|    _|  _|    _|  _|        _|          _|    _|_|    _|  _|            _|        _|    _|  _|        _|
    _|_|_|_|  _|    _|  _|  _|_|  _|  _|_|    _|    _|  _|  _|  _|  _|_|      _|_|_|    _|_|_|_|  _|        _|_|_|
    _|    _|  _|    _|  _|    _|  _|    _|    _|    _|    _|_|  _|    _|      _|        _|    _|  _|        _|
    _|    _|    _|_|      _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|        _|    _|    _|_|_|  _|_|_|_|

    A token is already saved on your machine. Run `huggingface-cli whoami` to get more information or `huggingface-cli logout` if you want to log out.
    Setting a new token will erase the existing one.
    To login, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tokens .
Token:
Add token as git credential? (Y/n) Y
Token is valid (permission: write).
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
```

```
7m 23s  completed at 3:34 AM
```



```
    A token is already saved on your machine. Run `huggingface-cli whoami` to get more information or `huggingface-cli logout` if you want to log out.
    Setting a new token will erase the existing one.
    To login, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tokens .
Token:
Add token as git credential? (Y/n) Y
Token is valid (permission: write).
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
Upload 2 LFS files: 100%  2/2 [04:04<00:00, 244.99s/it]
pytorch_model-00002-of-00002.bin: 100%  3.50G/3.50G [01:16<00:00, 40.6MB/s]
pytorch_model-00001-of-00002.bin: 100%  9.98G/9.98G [04:04<00:00, 25.7MB/s]
CommitInfo(commit_url='https://huggingface.co/aamirsea/Llama-2-7b-chat-finetune/commit/031c298bff0342306f4096a26e167a774e572695', commit_message='Upload
tokenizer', commit_description='', oid='031c298bff0342306f4096a26e167a774e572695', pr_url=None, pr_revision=None, pr_num=None)
```

https://huggingface.co/aamirsea/Llama-2-7b-chat-finetune/commit/031c298bff0342306f4096a26e167a774e572695

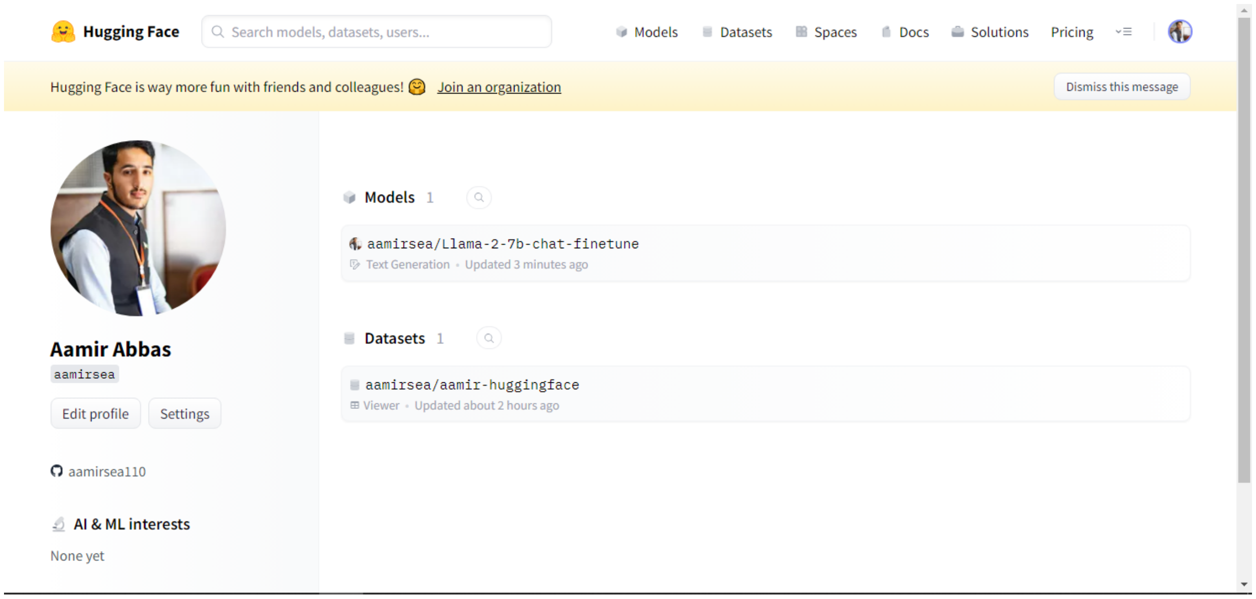


aamirsea / Llama-2-7b-chat-finetune

Llama-2-7b-chat-finetune

aamirsea Upload tokenizer 031c298 3 minutes ago

.gitattributes	1.52 kB	initial commit	10 minutes ago
config.json	630 Bytes	Upload LlamaForCausalLM	3 minutes ago
generation_config.json	183 Bytes	Upload LlamaForCausalLM	3 minutes ago
pytorch_model-00001-of-00002.bin	9.98 GB LFS	Upload LlamaForCausalLM	3 minutes ago
pytorch_model-00002-of-00002.bin	3.5 GB LFS	Upload LlamaForCausalLM	3 minutes ago
pytorch_model.bin.index.json	26.8 kB	Upload LlamaForCausalLM	3 minutes ago



Hugging Face is way more fun with friends and colleagues! Join an organization

Aamir Abbas

aamirsea

aamirsea110

AI & ML interests

None yet

Models 1

aamirsea/Llama-2-7b-chat-finetune

Text Generation · Updated 3 minutes ago

Datasets 1

aamirsea/aamir-huggingface

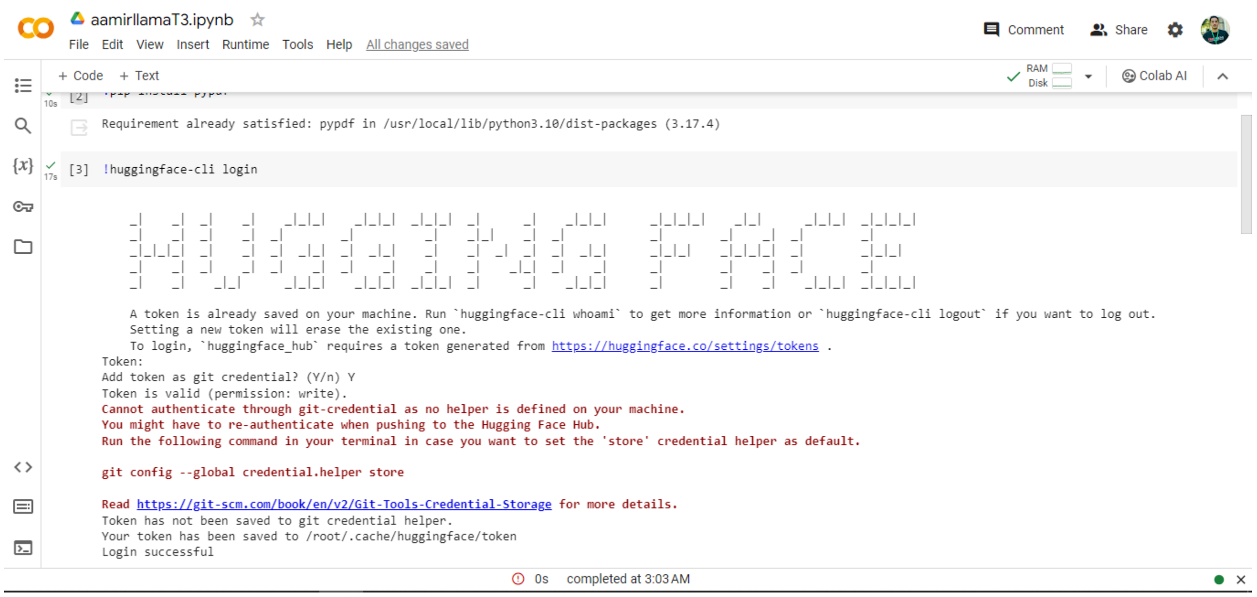
Viewer · Updated about 2 hours ago

Task 3

In this section, your objective is to leverage the fine-tuning model from Task 2 to construct a versatile chatbot utilizing LangChain.

Q3.1, Provide screenshots of the prompt template you have devised.

Q3.2, Provide the text generation outcomes achieved through your chatbot.

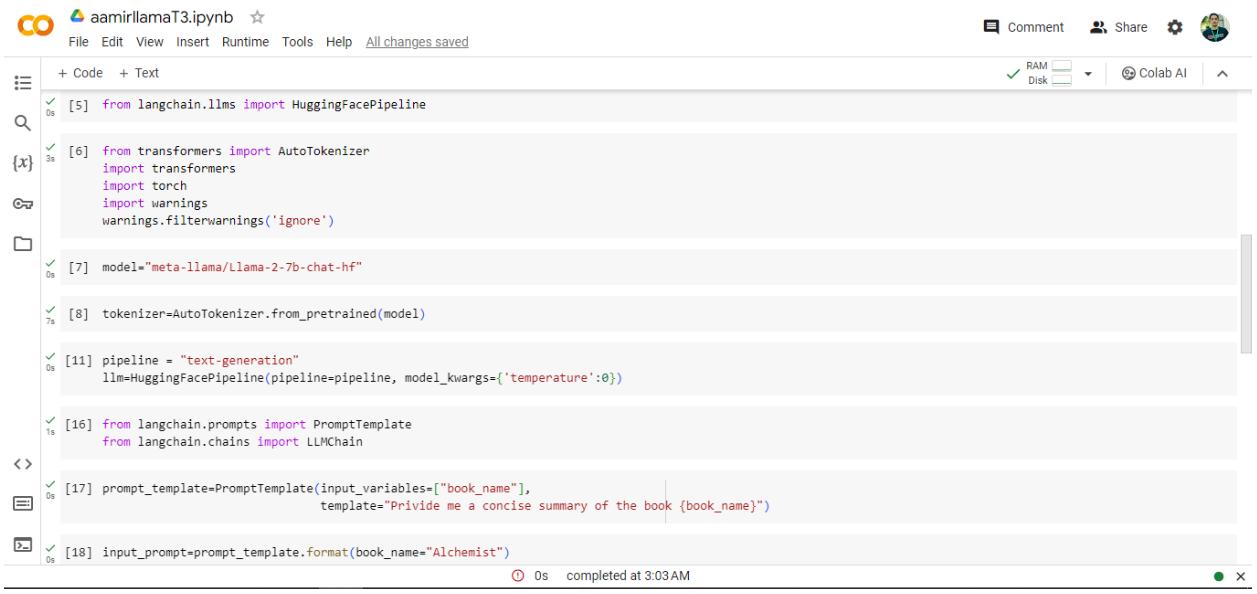


aamirllamaT3.ipynb

```
Requirement already satisfied: pypdf in /usr/local/lib/python3.10/dist-packages (3.17.4)
```

```
[3] !huggingface-cli login
```

```
    _|    _|  _|    _|    _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|_|_|_|    _|_|      _|_|_|  _|_|_|_|
    _|    _|  _|    _|  _|        _|          _|    _|_|    _|  _|            _|        _|    _|  _|        _|
    _|_|_|_|  _|    _|  _|  _|_|  _|  _|_|    _|    _|  _|  _|  _|  _|_|      _|_|_|    _|_|_|_|  _|        _|_|_|
    _|    _|  _|    _|  _|    _|  _|    _|    _|    _|    _|_|  _|    _|      _|        _|    _|  _|        _|
    _|    _|    _|_|      _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|        _|    _|    _|_|_|  _|_|_|_|

    A token is already saved on your machine. Run `huggingface-cli whoami` to get more information or `huggingface-cli logout` if you want to log out.
    Setting a new token will erase the existing one.
    To login, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tokens .
Token:
Add token as git credential? (Y/n) Y
Token is valid (permission: write).
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
```



aamirllamaT3.ipynb

```
[5] from langchain.llms import HuggingFacePipeline
```

```
[6] from transformers import AutoTokenizer
    import transformers
    import torch
    import warnings
    warnings.filterwarnings('ignore')
```

```
[7] model="meta-llama/Llama-2-7b-chat-hf"
```

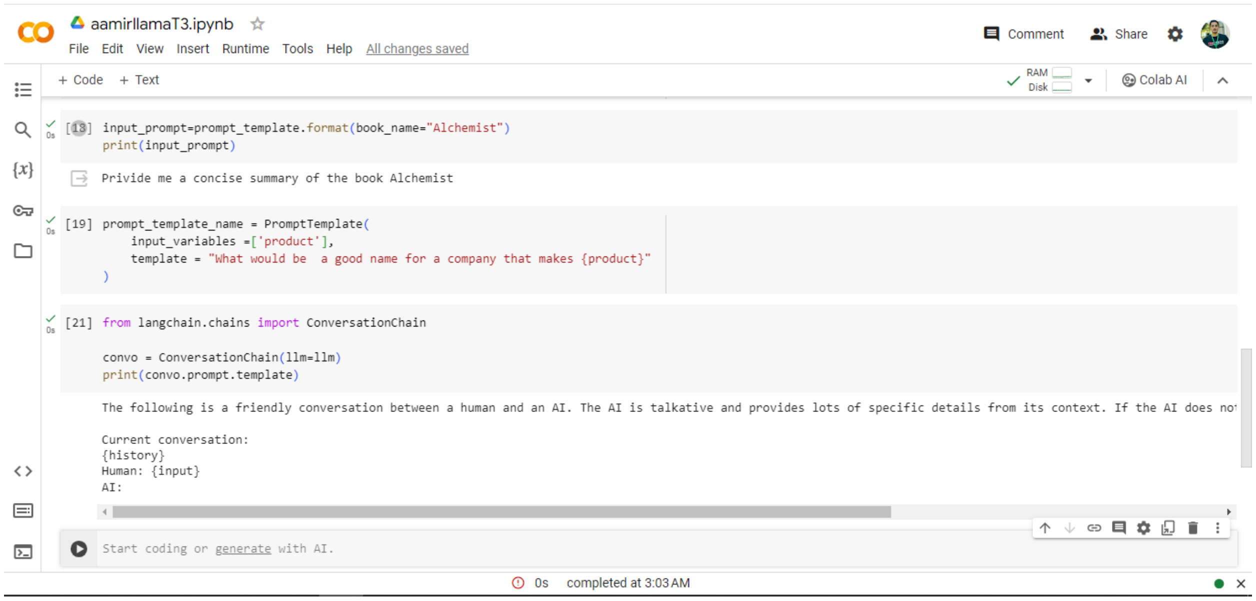
```
[8] tokenizer=AutoTokenizer.from_pretrained(model)
```

```
[11] pipeline = "text-generation"
     llm=HuggingFacePipeline(pipeline=pipeline, model_kwargs={'temperature':0})
```

```
[16] from langchain.prompts import PromptTemplate
     from langchain.chains import LLMChain
```

```
[17] prompt_template=PromptTemplate(input_variables=["book_name"],
                                    template="Privide me a concise summary of the book {book_name}")
```

```
[18] input_prompt=prompt_template.format(book_name="Alchemist")
```



```
[18] input_prompt=prompt_template.format(book_name="Alchemist")
     print(input_prompt)

     Privide me a concise summary of the book Alchemist

[19] prompt_template_name = PromptTemplate(
         input_variables =['product'],
         template = "What would be  a good name for a company that makes {product}"
     )

[21] from langchain.chains import ConversationChain

     convo = ConversationChain(llm=llm)
     print(convo.prompt.template)

     The following is a friendly conversation between a human and an AI. The AI is talkative and provides lots of specific details from its context. If the AI does not

     Current conversation:
     {history}
     Human: {input}
     AI:

     Start coding or generate with AI.
```