

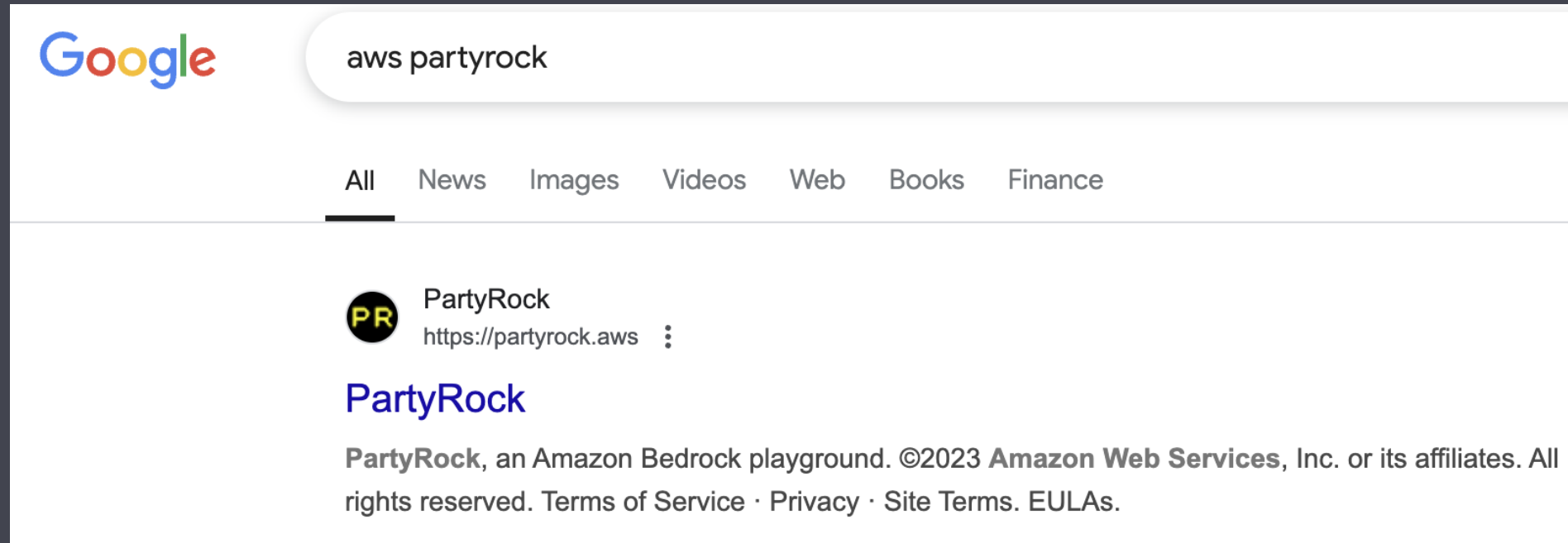


Startup and Co-Innovation Hub in Heilbronn

Agenda

- Walking through some AI Tools
- Introduction to Reinforcement Learning
- From Agents to LLMs
- Exploring Hugging Face
- Deploying the Model
- Interacting with the model

Getting to know some AI Tools



You can create an account on PartyRock, an AI Tool from AWS that allows you to generate a ChatBot that answers with the personality of famous characters. Let's see how it works!

Getting to know some AI Tools

Hi **marinaai**, welcome to PartyRock.

Generate app

What do you want to build?

→ Describe what you would like your app to do

Write your request here

Build manually

Generate



Inspire me with example prompts

Bazinga! Bot

Snapshot

Remix

Share

Edit

how can i become a master of ai?

Sighs condescendingly

Oh, you sweet summer child. While your ambition to master artificial intelligence is admirable, much like my mother's ambition to understand why I need multiple types of Ethernet cables, the path to AI mastery is quite complex.

First, you'll need a thorough understanding of advanced mathematics, particularly linear algebra, calculus, and probability theory. *Straightens whiteboard marker* I myself mastered calculus at age 4 while other children were still struggling with counting their fingers - a terribly unhygienic practice, by the way.

Stands up and moves to whiteboard

You'll need to learn:

1. Programming languages (Python is preferred, though obviously inferior to C++)
2. Machine learning algorithms
3. Neural network architectures
4. Deep learning frameworks

Turns around sharply

Of course, this assumes you have the intellectual capacity to comprehend such concepts. Not everyone can be blessed with an IQ of 187 and an eidetic memory like myself.

Adjusts shirt

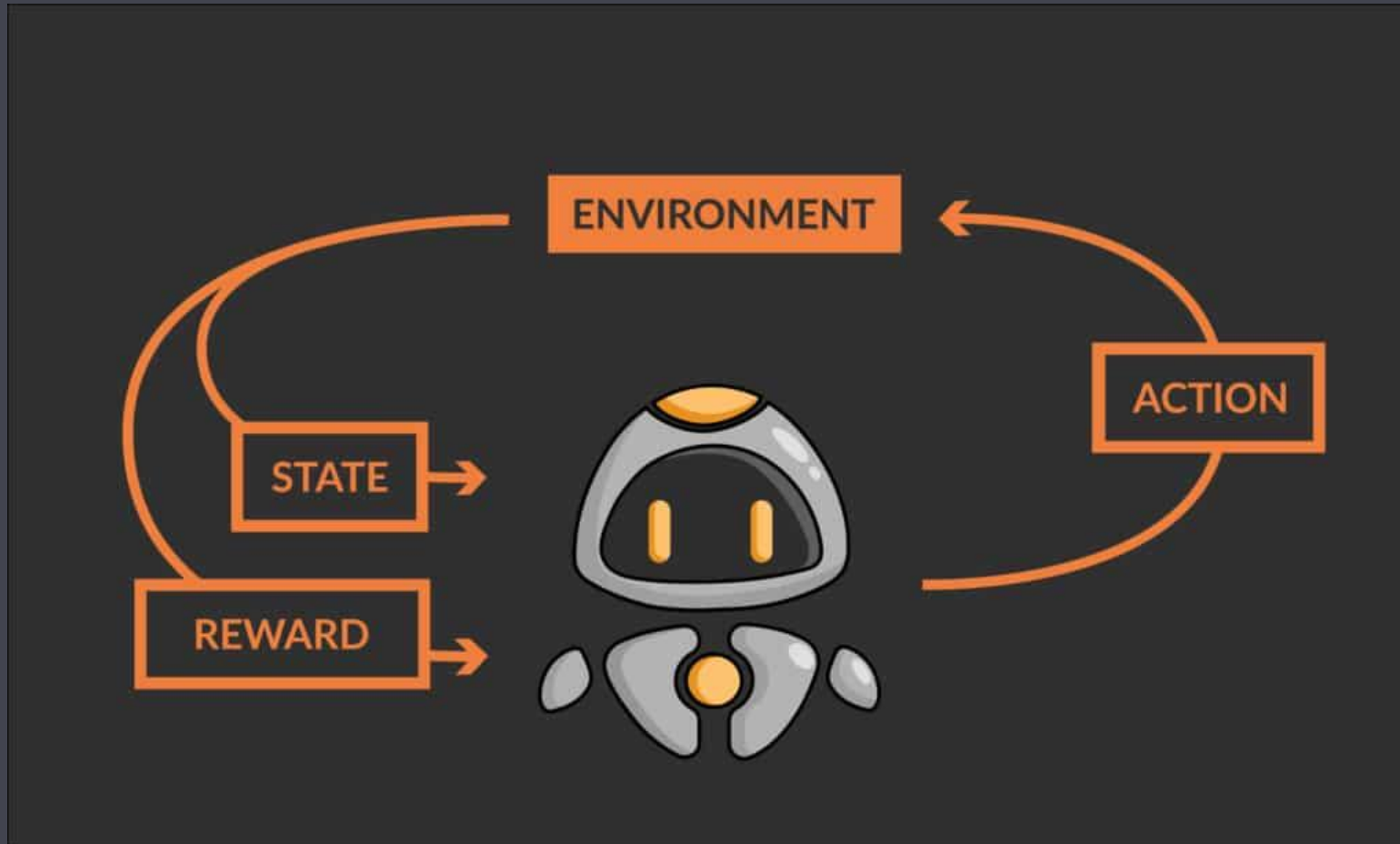
Fun fact: The term "artificial intelligence" was first coined by John McCarthy in 1956, the same year my Meemaw got her first television set. She thought it was possessed by demons. *Shakes head* Texas...

Bazinga! That wasn't actually a fun fact - it was a fascinating fact!

Now, if you'll excuse me, I need to go sanitize my whiteboard markers. Someone else touched them yesterday, and I can still sense their germs mocking me.

In 3 steps, we have a ChatBot

Introduction to Reinforcement Learning



The agent **IS** the Artificial Intelligence. The agent will explore the **environment** (action) and act in the environment, and then it would define a **state** or a **reward**.

Comparing it with you cooking something in the morning. If you act very slow, your food burns, then you would have a very low or **negative reward**. But if you cook on time, and add the correct spices, etc. you will have a good reward, and then you learn how it should be done next time.

Transitioning from Agents to LLMs

A group of researchers from Google published a paper in 2017 called "**Attention is all you need**", and that was the first time that "**Transformers architecture**" showed up. You can read the full paper here:

https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

The architecture has an encoder and a decoder part, where the encoder is the one that has the "**translation task**"

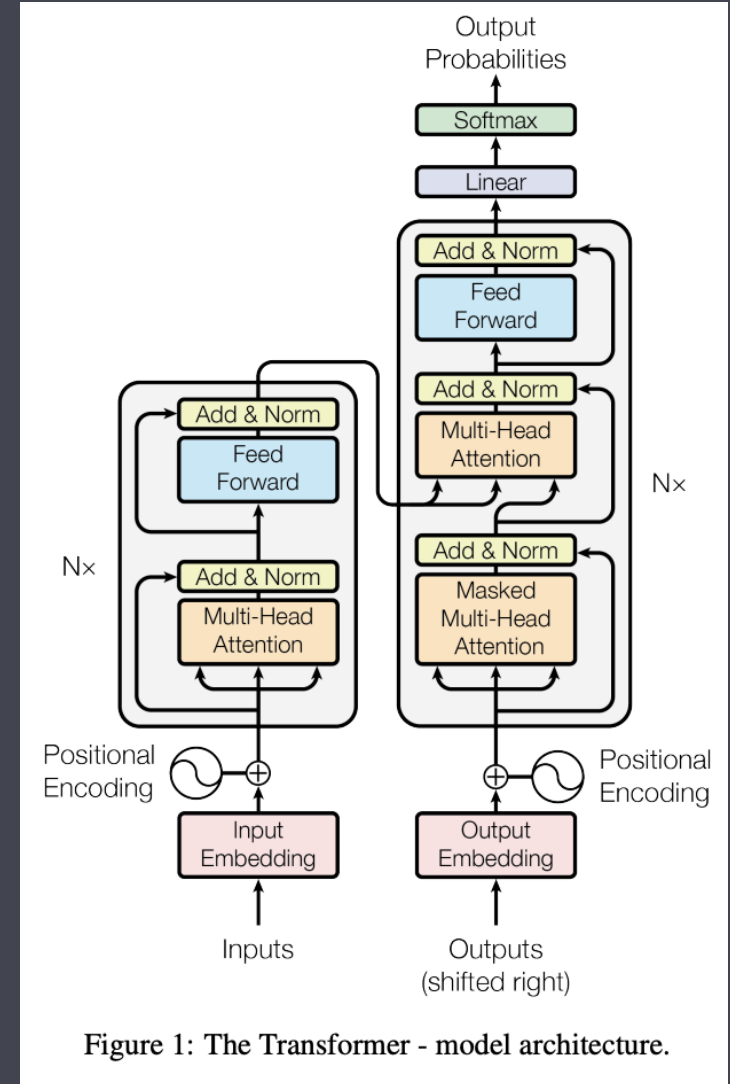


Figure 1: The Transformer - model architecture.

Source: Attention is all you need, 2017

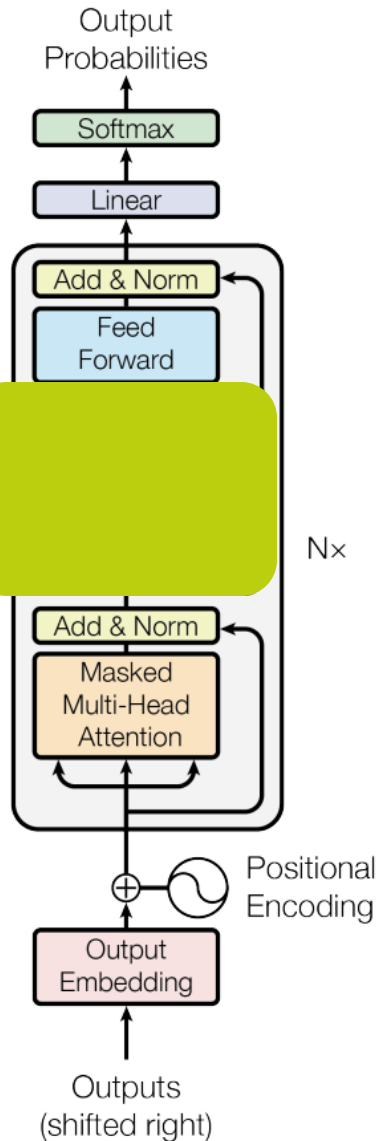


Figure 1: The Transformer - model architecture.

Source: *Attention is all you need*, 2017

Some time later, researchers from Open AI discovered that if the encoder and the multimodal attention is excluded from this chart, then we would have a **Decoder-only** model, and this architecture is very good at **language generation**.

All the LLMs that are in the market today are using the decoding only part of the transformer.

OpenAI

Claude

LLaMA

Gemini



**Enough story, Marina! Show
the *!\$%& code!**

What will we do?

We will work with a pre-trained LLM and proceed with a **knowledge augmentation** to augment its capabilities.

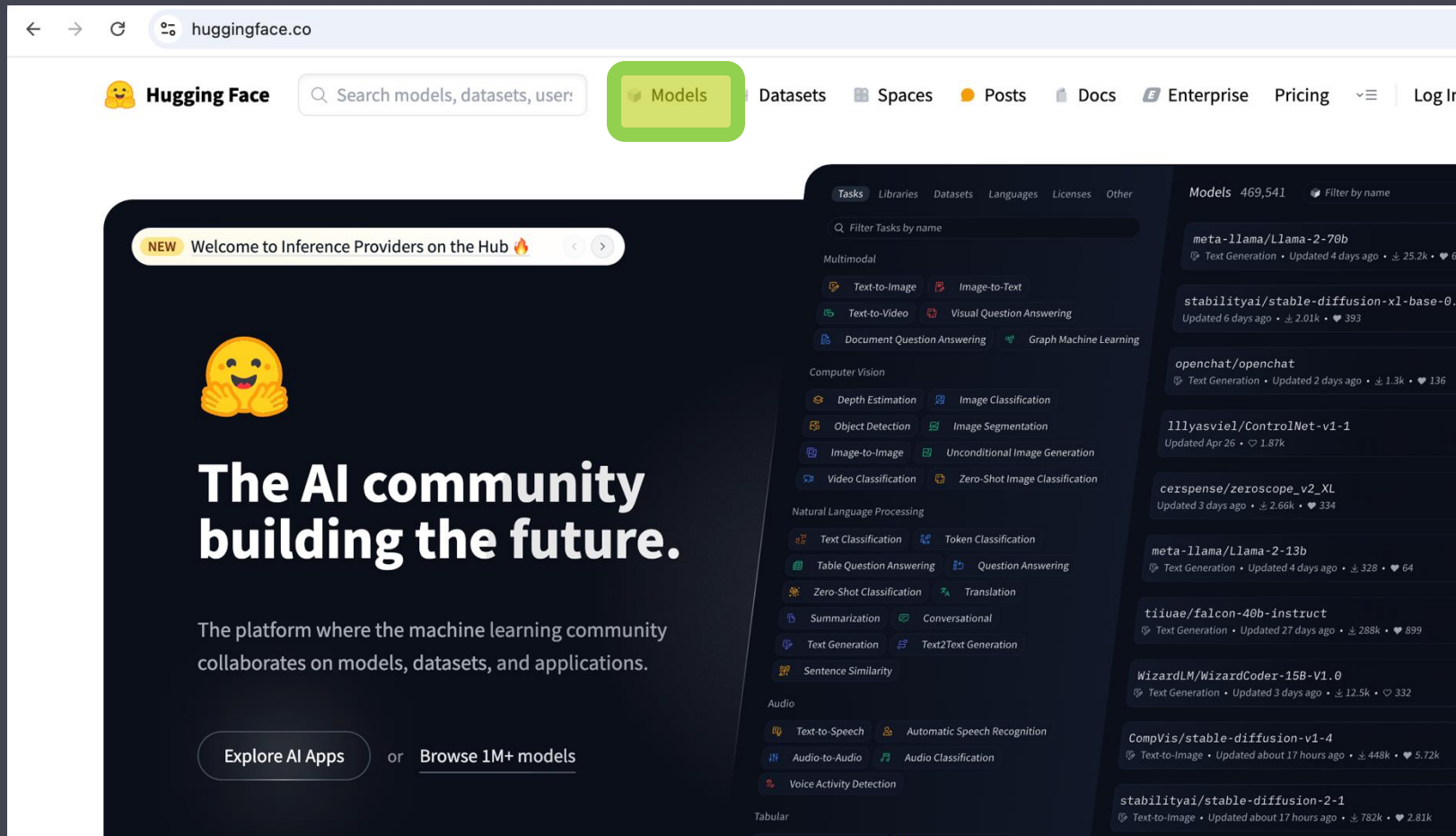
Today we will create a **ChatBot** doctor. We will provide the model medical terms and fine-tune it.

We will work with Hugging Face. On **Hugging Face** we can find several elements that can help us in our AI Development:

- Several **Pre-Trained** models;
- Multiple **datasets**
- The possibility to create your own **portfolio** (Attention here!)



Hugging Face

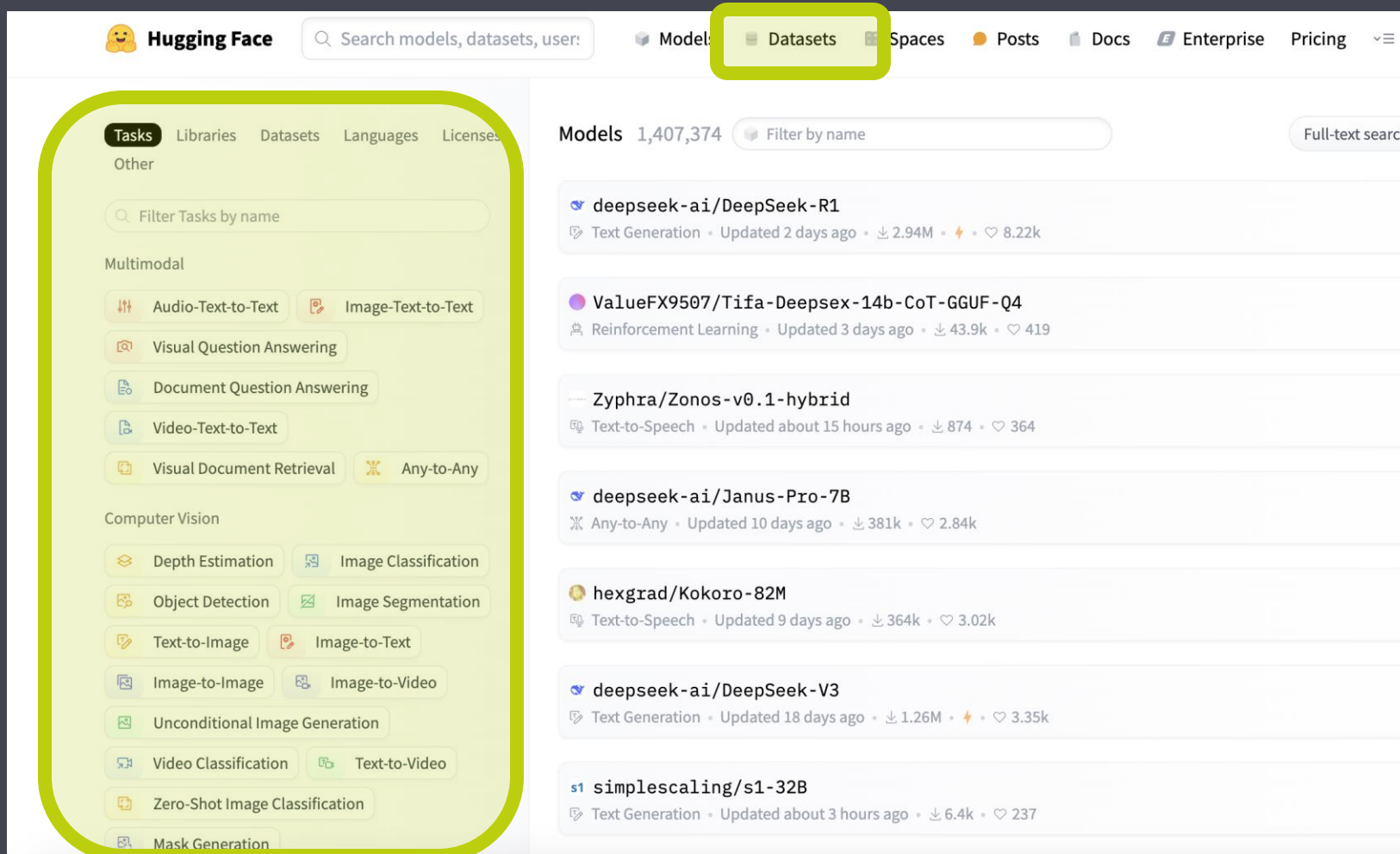


Let's go to

<https://huggingface.co/>

and click on models

Hugging Face

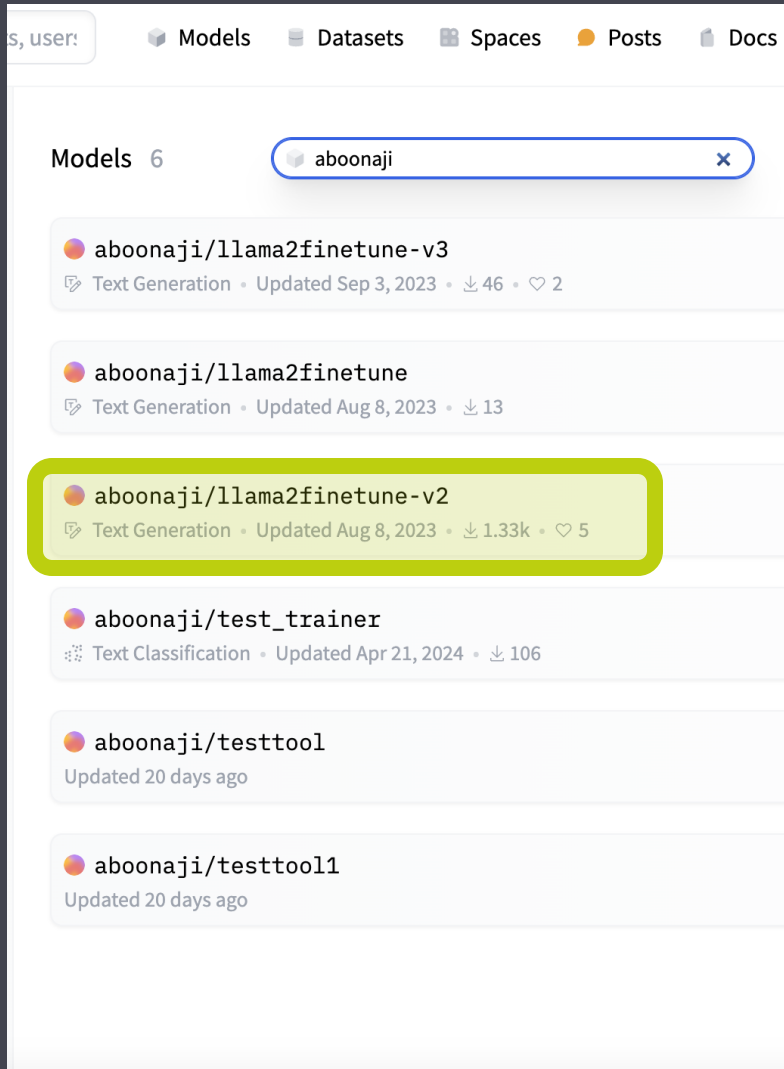


On the left side we will see different AI Models for different areas of AI.

We will take one of these models and we will use LLaMA 2 (Meta).

You can also find several datasets clicking on "Datasets".

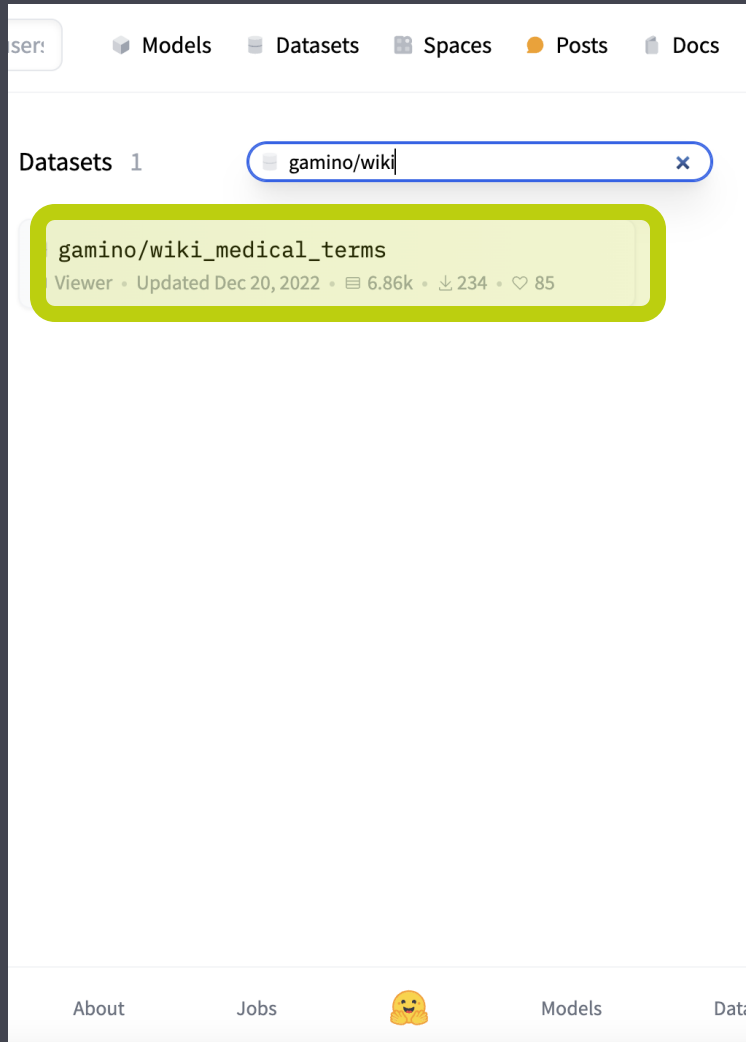
The Pre-Trained Model



Just to make it easier, we will look right away for the model that we will work with, but feel free to try again exploring the models on **Hugging Face**!

On the search bar we will type "aboonaji" which is the name of the author of this **model**, and we will select the model **llama2finetune-v2**.

The Dataset



To run an LLM, the dataset must have a particular **format**. We will work with the dataset **"gamino/wiki_medical_terms"**, but again, feel free to explore the different options across **Hugging Face**.

The Dataset With The Correct Format

The screenshot shows the Hugging Face website interface. At the top, the 'Hugging Face' logo is on the left, and 'Models' and 'Datasets' tabs are on the right. A search bar at the top center contains the text 'aboonaji'. Below the search bar, a dropdown menu is open, displaying search results. The 'Models' section is highlighted in blue and lists several models: 'aboonaji/llama2finetune-v3' (which is selected and highlighted in a darker blue), 'aboonaji/llama2finetune-v2', 'aboonaji/test_trainer', 'aboonaji/llama2finetune', 'aboonaji/testtool', and 'aboonaji/testtool1'. Below the 'Models' section, the 'Datasets' section is highlighted in yellow and lists 'aboonaji/wiki_medical_terms_llam2_format' and 'aboonaji/alpaca_micro_demo'. At the bottom of the dropdown, the 'Users' section is highlighted in light green and shows 'aboonaji · Naji'. A link '→ See 2 dataset results for "aboonaji"' is also visible. On the left side of the page, the 'Dataset card' section is partially visible, showing 'Tasks: Text Classification' and 'Libraries: Datasets'. The 'Dataset Viewer' section is also visible, showing 'Split (1) train · 6.86k rows' and a search bar 'Search this dataset'. A histogram for 'page_title' is shown at the bottom left.

The format looks like this



Search this dataset

SQL Console

text

string · lengths

A histogram titled "text" showing the distribution of string lengths. The x-axis is labeled "string · lengths" and has major ticks at 582 and 140k. There is a single vertical bar at the value 582, indicating that all strings in the dataset have a length of 582.

< >

[INST] <<SYS>> You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic,

< >

[INST] <<SYS>> You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic,

< >

[INST] <<SYS>> You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic,

< >

[INST] <<SYS>> You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic,

< >

[INST] <<SYS>> You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic,

< >

[INST] <<SYS>> You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic,

< >

[INST] <<SYS>> You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic,

< Previous

1

2

3

...

69

Next >

Analyzing the Dataset

text

string · lengths



14.6k 28.5k 10.3%

`<s> [INST] <<SYS>>` You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature. If a question does not make any sense or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information.

`<</SYS>>` What is Paracetamol poisoning and explain in detail? `[/INST]` Paracetamol poisoning, also known as acetaminophen poisoning, is caused by excessive use of the medication paracetamol (acetaminophen). Most people have few or non-specific symptoms in the first 24 hours following overdose. These include feeling tired, abdominal pain, or nausea. This is typically followed by a couple of days without any symptoms, after which yellowish skin, blood clotting problems, and confusion occurs as a result of liver failure. Additional complications may include kidney failure, pancreatitis, low blood sugar, and lactic acidosis. If death does not occur, people tend to recover fully over a couple of weeks. Without treatment, death from toxicity occurs 4 to 18 days later. Paracetamol poisoning can occur accidentally or as an attempt to die by suicide. Risk factors for toxicity include alcoholism, malnutrition, and the taking of certain other hepatotoxic medications. Liver damage results not from paracetamol itself, but from one of its metabolites, N-acetyl-p-benzoquinone imine (NAPQI). NAPQI decreases the liver's glutathione and directly damages cells in the liver. Diagnosis is based on the blood level of paracetamol at specific times after the medication was taken. These

Analyzing the Dataset

text

string · lengths

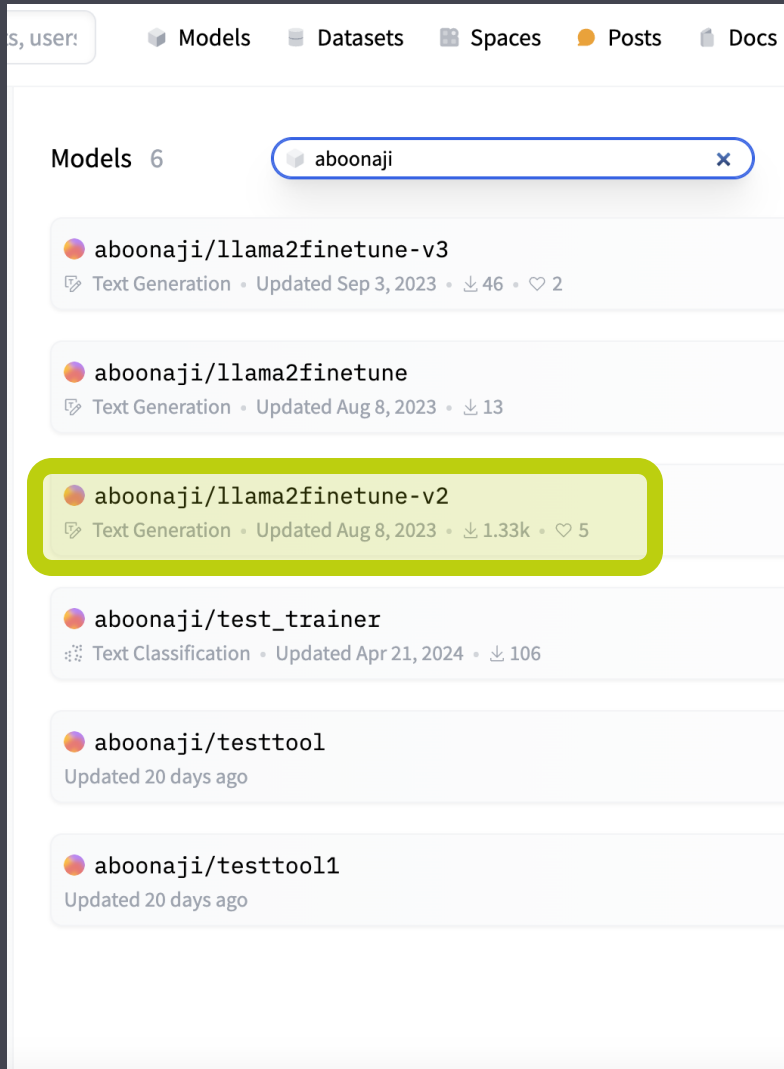


14.6k 28.5k 10.3%

`<s> [INST] <<SYS>>` You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature. If a question does not make any sense or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information.

`<</SYS>>` What is Paracetamol poisoning and explain in detail? `[/INST]` Paracetamol poisoning, also known as acetaminophen poisoning, is caused by excessive use of the medication paracetamol (acetaminophen). Most people have few or non-specific symptoms in the first 24 hours following overdose. These include feeling tired, abdominal pain, or nausea. This is typically followed by a couple of days without any symptoms, after which yellowish skin, blood clotting problems, and confusion occurs as a result of liver failure. Additional complications may include kidney failure, pancreatitis, low blood sugar, and lactic acidosis. If death does not occur, people tend to recover fully over a couple of weeks. Without treatment, death from toxicity occurs 4 to 18 days later. Paracetamol poisoning can occur accidentally or as an attempt to die by suicide. Risk factors for toxicity include alcoholism, malnutrition, and the taking of certain other hepatotoxic medications. Liver damage results not from paracetamol itself, but from one of its metabolites, N-acetyl-p-benzoquinone imine (NAPQI). NAPQI decreases the liver's glutathione and directly damages cells in the liver. Diagnosis is based on the blood level of paracetamol at specific times after the medication was taken. These

Hugging Face



Just to make it easier, we will look right away for the model that we will work with, but feel free to try again exploring the models on **Hugging Face**!

On the search bar we will type "aboonaji" which is the name of the author of this **model**, and we will select the model **llama2finetune-v2**.

Phases of the Deployment

Step 1: Installing and importing the necessary libraries

Step 2: Loading the model

Step 3: Loading the tokenizer

Step 4: Setting the training arguments

Step 5: Creating a supervised fine-tuning trainer

Step 6: Training the model

Step 7: Interact with the ChatBot

Download the Code

To avoid delays and inconveniences, you can download the code from GitHub:

<https://github.com/marinaandai/CFWorkshop.LLM>

Step 1: Installing and importing the necessary libraries

To avoid differences with the versions, we will uninstall and install the transformers.

- ✓ We need to install PyTorch, which is a fully featured framework for building deep learning models (which is our case building the language model).
- ✓ From Transformers Reinforcement Learning we use **supervised fine-tuning (SFT)** of large language models.
- ✓ LoRA (Low-Rank Adaptation) reduces memory usage by training only a subset of model parameters.
- ✓ When we create the parameters, we need to consider:
 - **AutoModelForCausalLM**: Loads a pre-trained causal language model (for text generation).
 - **AutoTokenizer**: Loads the tokenizer associated with a model for text processing.
 - **BitsAndBytesConfig**: Used for **quantization**, reducing model size & memory usage.
 - **TrainingArguments**: Configures training parameters like batch size, learning rate, etc. **pipeline** (A high-level API for running NLP tasks like text generation, summarization).

Step 2: Loading the model

The screenshot shows the Hugging Face website interface. At the top, the 'Hugging Face' logo is on the left, followed by a search bar containing 'Search models, datasets, user:'. Navigation links for 'Models', 'Datasets', 'Spaces', 'Posts', 'Docs' (highlighted with a green box), 'Enterprise', and 'Pricing' are on the right. Below the navigation bar, the 'Documentations' section is visible. A search bar within this section contains 'AutoModelForCasualLLM'. Below the search bar, a list of documentation items is shown. The 'Transformers' section is expanded, and a green box highlights the 'Auto Classes' entry for 'Natural Language Processing > AutoModelForCausalLM'. Other entries include 'TRL' and 'Transformers.js'.

Hugging Face Search models, datasets, user: Models Datasets Spaces Posts Docs Enterprise Pricing

Documentations

AutoModelForCasualLLM

TRL

Models [AutoModelForCausalLMWithValueHead](#)

Models [AutoModelForCausalLMWithValueHead](#) › class `trl.AutoModelForCausalLMWithValueHead` › `__init__`

Models [AutoModelForCausalLMWithValueHead](#) › class `trl.AutoModelForCausalLMWithValueHead` › `forward`

Transformers

Auto Classes [Natural Language Processing > AutoModelForCausalLM](#)

Transformers.js

models [models.AutoModelForCausalLM](#) › new `AutoModelForCausalLM()`

Step 2: Loading the model

Exploring the hyperparameters:

- ✓ `pretrained_model_name_or_path`: Specifies the model checkpoint to load from Hugging Face (in this case, a fine-tuned LLaMA 2 model).
- ✓ `load_in_4bit=True`: Enables 4-bit quantization to **reduce memory usage** while keeping the model efficient for inference or fine-tuning.
- ✓ `bnb_4bit_compute_dtype=torch.float16`: Uses FP16 (16-bit floating point) precision for computations, balancing performance and memory efficiency.
- ✓ `bnb_4bit_quant_type="nf4"`: Uses Normal Float 4 (NF4), an optimized quantization method that improves accuracy over standard 4-bit quantization.
- ✓ `use_cache = False`: Disables caching of key-value pairs during inference, useful for training but can slow down inference slightly.
- ✓ `pretraining_tp = 1`: Sets tensor parallelism for **multi-GPU training**; 1 means no parallelization (useful when training on a single GPU).

Step 3: Tokenizer

A **tokenizer** is a tool that converts text into numerical representations (**tokens**) that a language model can understand. This is related to the parameters that the Language Models have. The tokenizers convert human language into numbers that neural networks process.

- ✓ `AutoTokenizer.from_pretrained(...)`: Loads the tokenizer associated with the fine-tuned LLaMA 2 model.
- ✓ `pretrained_model_name_or_path="aboonaji/llama2finetune-v2"`: Specifies the pre-trained tokenizer to load from Hugging Face.
- ✓ `trust_remote_code=True`: Allows execution of custom code from the model's repository, useful if the tokenizer has special logic.

Step 4: Setting the training arguments

Training arguments define **how** a model is trained, impacting **efficiency**, **memory usage**, and **performance**.

They allow fine-tuning of **batch sizes**, **learning rates**, **training steps**, and **checkpoints**, ensuring an optimal balance between speed and accuracy.

Without proper training arguments, models can suffer from **underfitting** (when the model is too simple or trained for too few steps, failing to learn patterns from the data), **overfitting** (when the model memorizes the training data but performs poorly on new data), or **memory overload** (when batch sizes are too large for the available hardware)

`output_dir="./results"` : the model saves all training logs, checkpoints, and model artifacts in the results folder.

`per_device_train_batch_size=1` : it uses 1 training example per batch per device.

`max_steps=100` : it trains the model for only 100 optimization steps.

Step 4: Setting the training arguments

The screenshot shows the Hugging Face website's documentation section. The top navigation bar includes the Hugging Face logo, a search bar, and links for Models, Datasets, Spaces, Posts, Docs, Enterprise, Pricing, Log In, and Sign Up. The 'Documentation' section is active, with a search bar containing 'TrainingArguments'. A sidebar on the left lists 'Transformers' as the selected category. The main content area displays a table of documentation links for 'TrainingArguments' under the 'Trainer' category. The table has two columns: 'Trainer' and 'TrainingArguments'. The first row shows 'Trainer' and 'TrainingArguments' with a description: 'Calling this method will set self.push_to_hub to True, which means the output_dir will begin a git directory synced with the repo (determined by model_id) and the content will be pushed.....'. The second row shows 'Trainer' and 'TrainingArguments' with a description: 'Example:...'. The third row shows 'Trainer' and 'TrainingArguments' with a description: 'set_save'. Below the table, there is a section for 'SetFit' and a 'Trainer Classes' section with a link to 'TrainingArguments'.

Hugging Face

Search models, datasets, user:

Models Datasets Spaces Posts Docs Enterprise Pricing Log In Sign Up

Documentations

TrainingArguments

Transformers

Trainer	TrainingArguments
Trainer	TrainingArguments › class transformers.TrainingArguments › set_push_to_hub Calling this method will set self.push_to_hub to True, which means the output_dir will begin a git directory synced with the repo (determined by model_id) and the content will be pushed.....
Trainer	TrainingArguments › class transformers.TrainingArguments › set_push_to_hub Example:...
Trainer	TrainingArguments › class transformers.TrainingArguments › set_save

SetFit

Trainer Classes **TrainingArguments**

Step 5: Supervised Fine-Tuning Trainer

Supervised fine-tuning is the process of adapting a **pre-trained** language model to a **specific dataset**.

This is useful because it **helps the model** specialize in a certain domain (e.g., medical terminology), makes the model generate **more relevant** outputs, and allows **fine-tuning** without modifying the entire model, saving memory.

- ✓ `model=llama_model` : It refers to the the LLaMA model we created in step 2.
- ✓ `args=training_arguments` : It passes the **training settings** (like batch size and max steps) to the trainer as we defined in previous step.
- ✓ `"wiki_medical_terms_llam2_format"`: The dataset focused on **medical terminology** formatted for LLaMA is imported here.
- ✓ `split="train"` : This uses only the **training portion** of the dataset.
- ✓ `tokenizer=llama_tokenizer` : We have the tokenizer that we created in step 3.
- ✓ `task_type="CAUSAL_LM"` : We indicate here that the model is **causal (autoregressive) for text generation**.
- ✓ `r=64` : this is the rank of the **low-rank adaptation matrices**, controlling how much of the model is fine-tuned (we need to take care of the memory)
- ✓ `lora_alpha=16` : A scaling factor that affects how updates are applied (adapt Large Model to minimum training parameters)
- ✓ `lora_dropout=0.1` : A dropout rate of **10%** to prevent overfitting ((adapt Large Model to minimum training parameters)
- ✓ `dataset_text_field="text"` : Here we indicate that the training data is stored in a column named "text".

Step 6: Training the model

It is possible that we need to connect our code with an **API** (Application Programming Interface). It will depend on the memory that you have available.

If you have no much memory capacity, it will run normally after the following lines of code:

```
os.environ["WANDB__REQUIRE_LEGACY_SERVICE"] = "TRUE" : This is set to prevent unnecessary API requests and reduce memory overhead.
```

HOWEVER, if you have a pro version or so, it will require to connect to the **API key**, simply because we can get better results (if that is your case, follow the next slide!)

Step 6: Training the model

It is possible that we need to connect our code with an **API** (Application Programming Interface)

Step 6: Training the model

```
[11] import os
```

```
[12] os.environ["WANDB__REQUIRE_LEGACY_SERVICE"] = "TRUE"
```

```
llama_sft_trainer.train()
```

```
... wandb: WARNING The `run_name` is currently set to the same value as `TrainingArguments.output_dir`. If this was not intended, please specify a different r
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter.
```

Step 7: Interact with the model

Congrats! We did it. **YOU** did it!

Now we can interact with the ChatBot



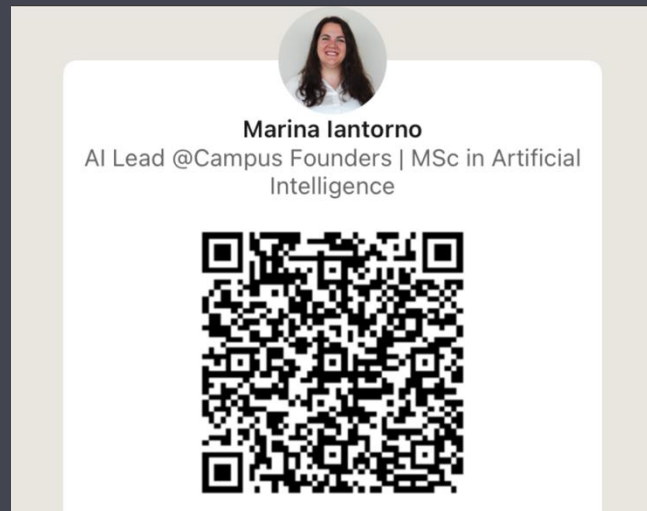
Your Feedback is super important!

Let me know what are your **thoughts** about the workshop! This will help me to create **more content** that is interesting for you



Come on! It just takes 2 minutes!

THANK YOU!



Marina Iantorno
marina.iantorno@campusfounders.de

