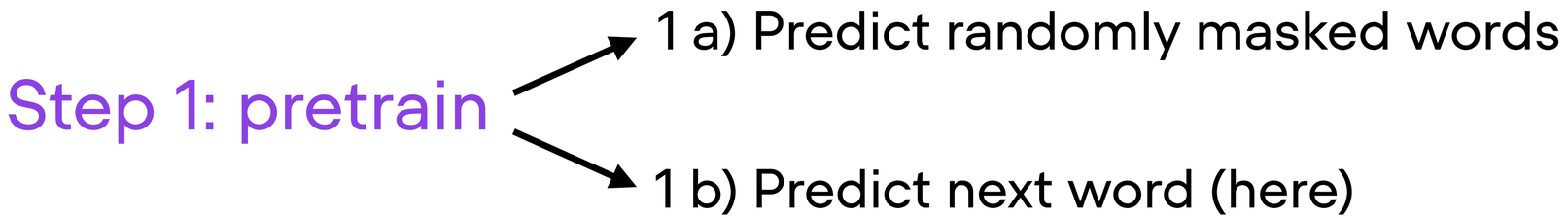
8.6

Large Language Models

Part 2: Generative Pretrained Transformer (GPT)

Sebastian Raschka and the Lightning AI Team

Self-supervised pretraining



Step 2: fine-tune

	Date
GPT 1	2018
GPT 2	2019
GPT 3	2020
InstructGPT & ChatGPT	2022

	Date	Size
GPT 1	2018	110 million
GPT 2	2019	1.5 billion
GPT 3	2020	175 billion
InstructGPT & ChatGPT	2022	GPT 3-based

	Date	Size	Paper
GPT 1	2018	110 million	Improving Language Understanding by Generative Pre-Training
GPT 2	2019	1.5 billion	Language Models are Unsupervised Multitask Learners
GPT 3	2020	175 billion	Language Models are Few-Shot Learners
InstructGPT & ChatGPT	2022	**GPT 3-based**	Aligning Language Models to Follow Instructions

GPT models are generative models (as opposed to classifiers)

GPT models are generative models (as opposed to classifiers)

How to make a banana smoothie?

To make a banana smoothie, you will need the following ingredients:

- 1 ripe banana
- 1 cup of milk (dairy or non-dairy)
- 1/2 cup of plain yogurt (dairy or non-dairy)
- 1 tablespoon of honey (optional)

Instructions:

1. Peel the banana and cut it into small chunks.
2. Add the banana chunks, milk, yogurt, and honey (if using) to a blender.
3. Blend the ingredients on high speed until smooth.
4. Taste and adjust sweetness if necessary.
5. Pour the smoothie into a glass and enjoy!

   You can also add some ice cubes, or some other fruits of your choice.

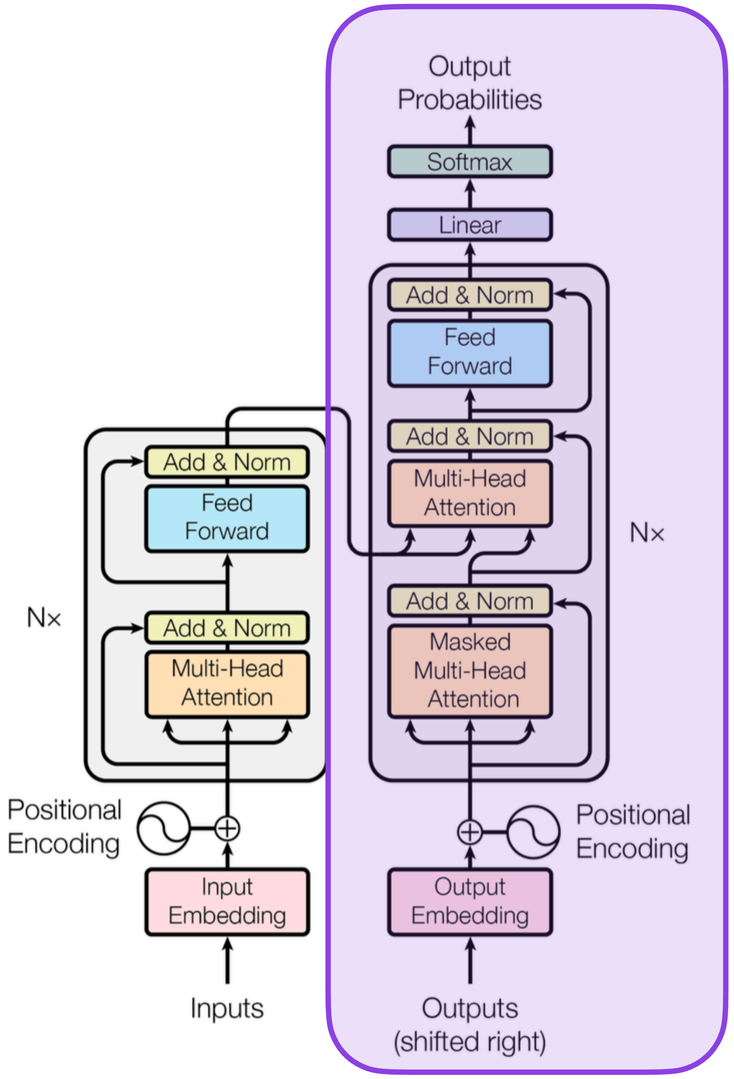
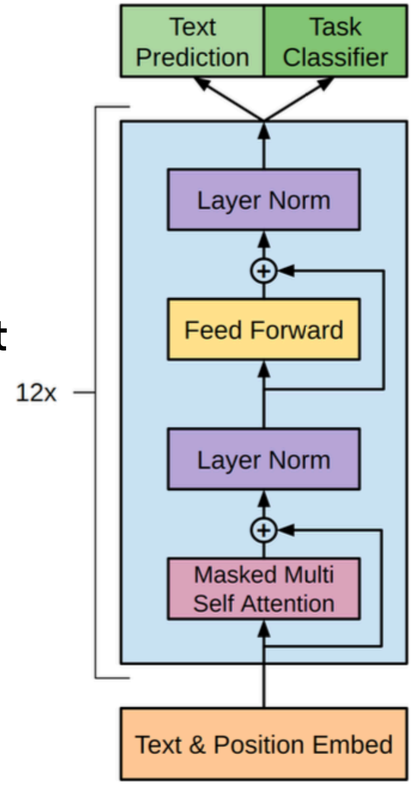


Figure 1: The Transformer - model architecture.

GPT is essentially the decoder part of the original transformer



https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

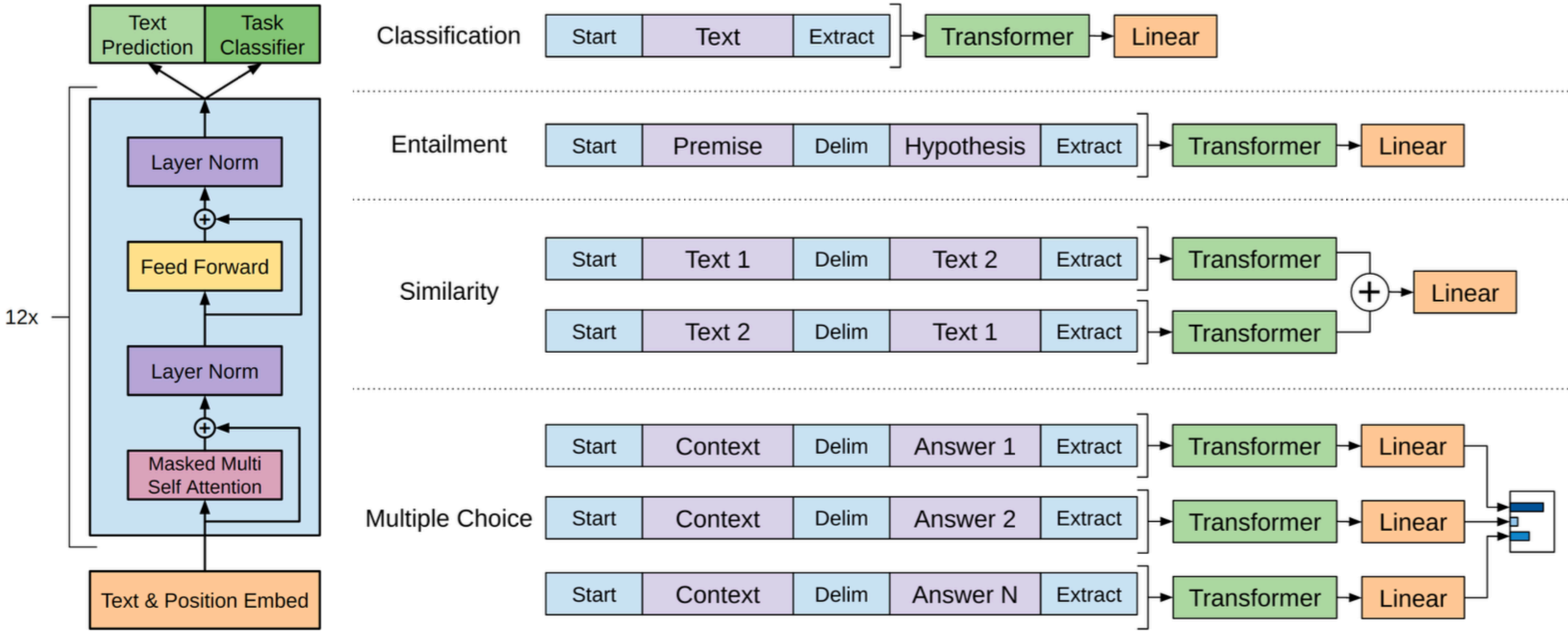
Feed model text from left to right, and it learns to predict the next word.

Self-supervised pretraining

Step 1: pretrain ⟶ Predict next word

Step 2: fine-tune

Fine-tune for target task

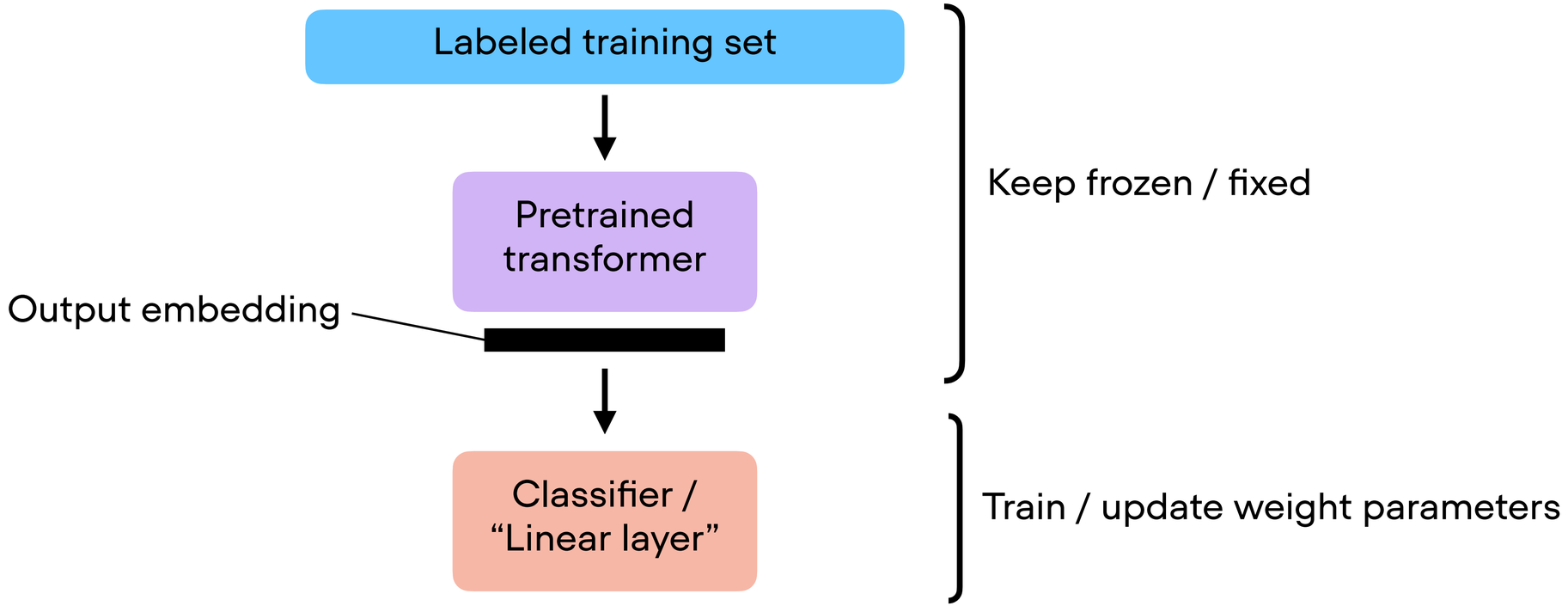


2 ways of adopting a pretrained transformer

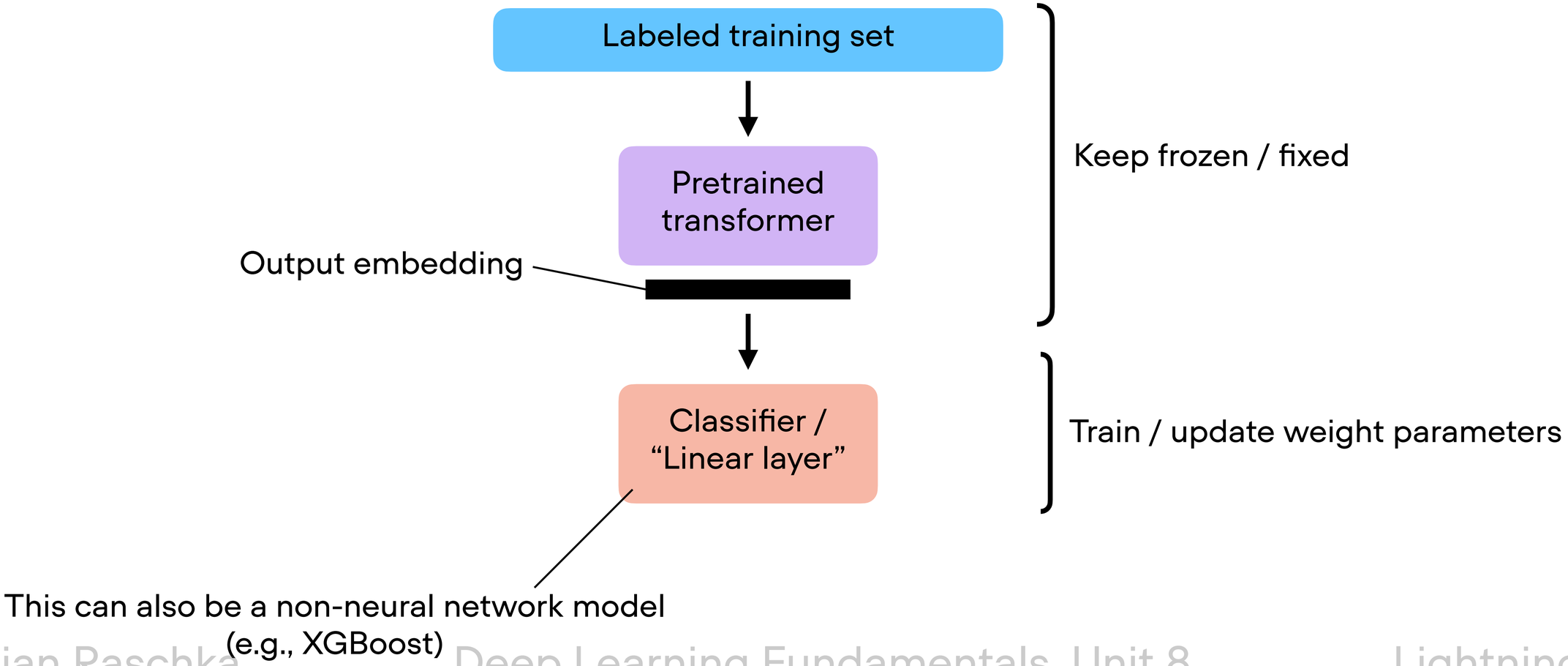
1) Feature-based approach

2) Fine-tuning approach

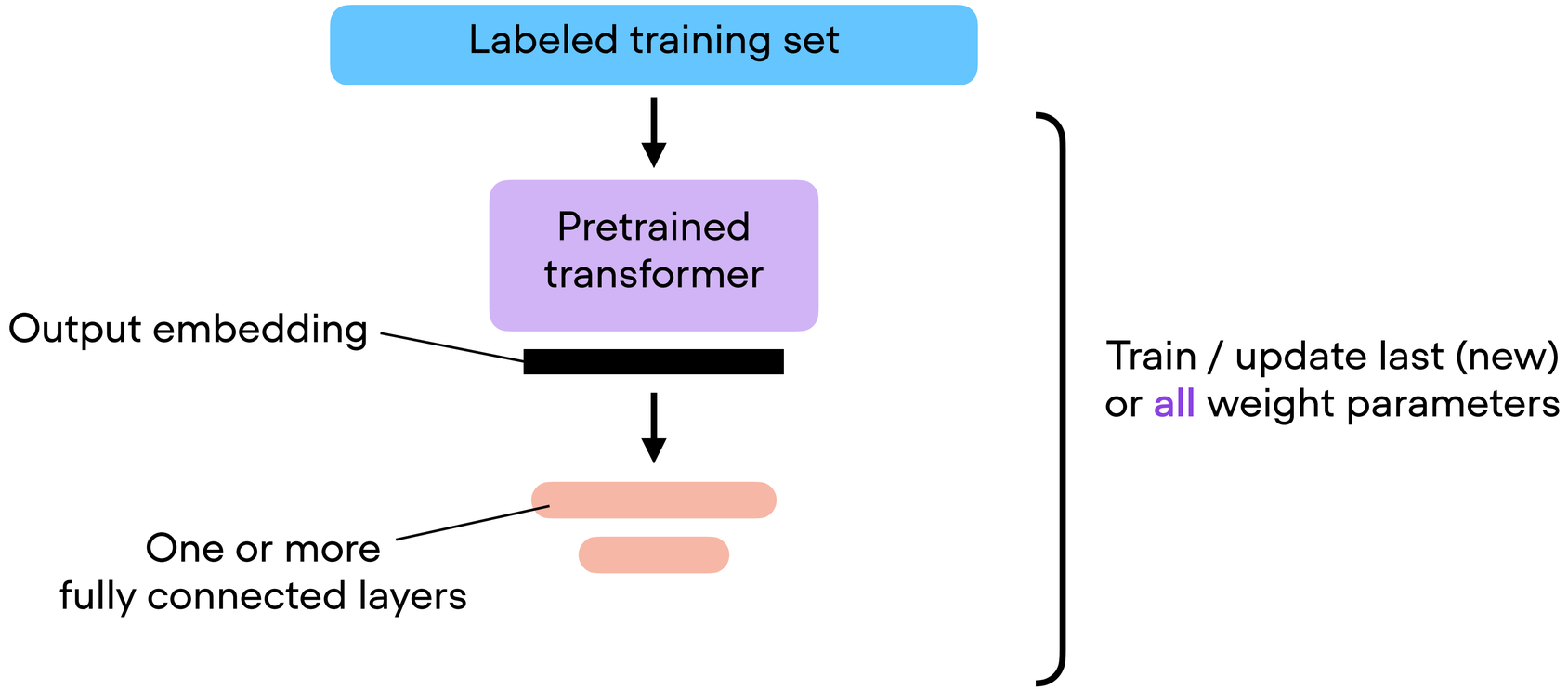
1) Feature-based approach



1) Feature-based approach

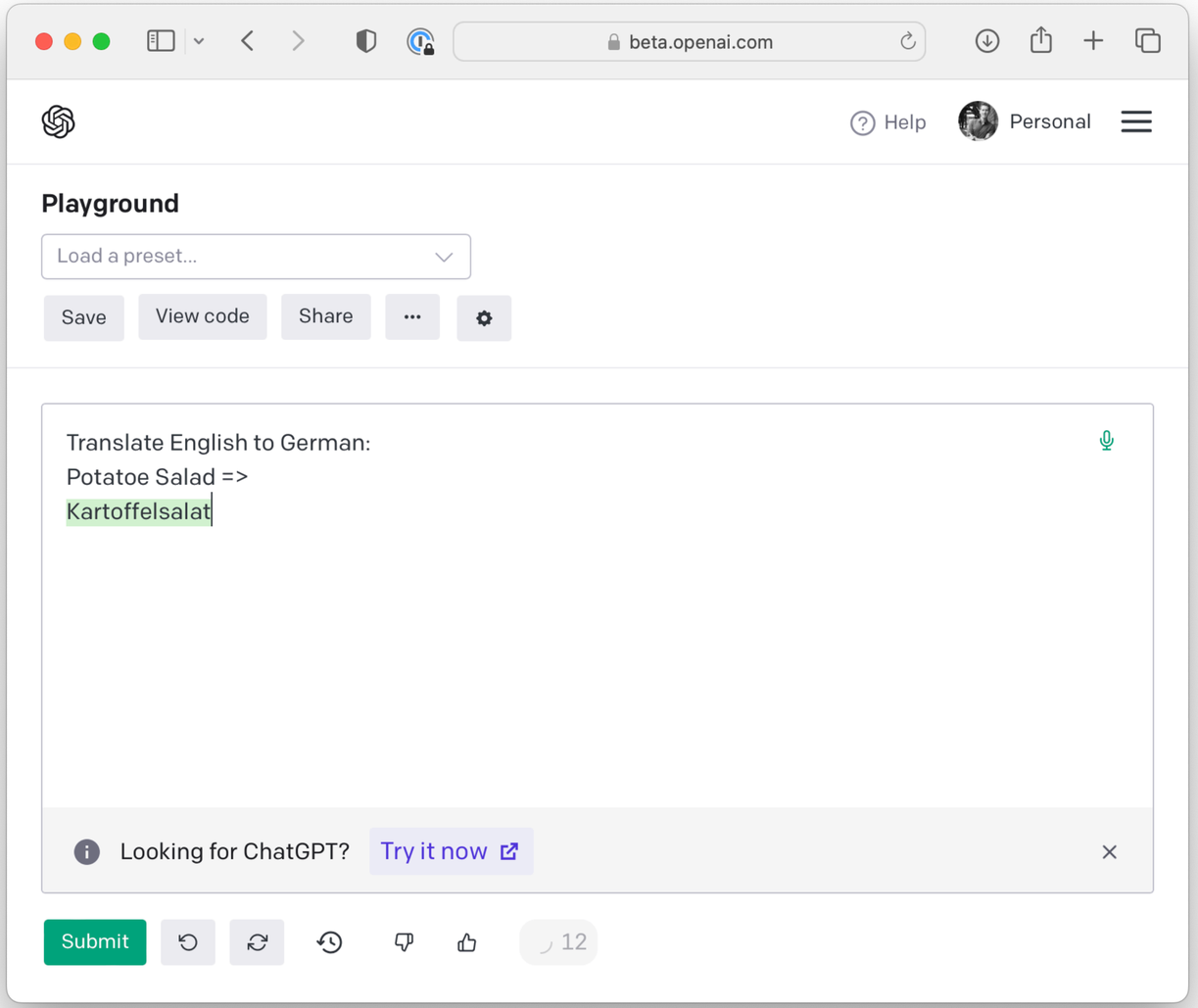


1) Fine-tuning approach

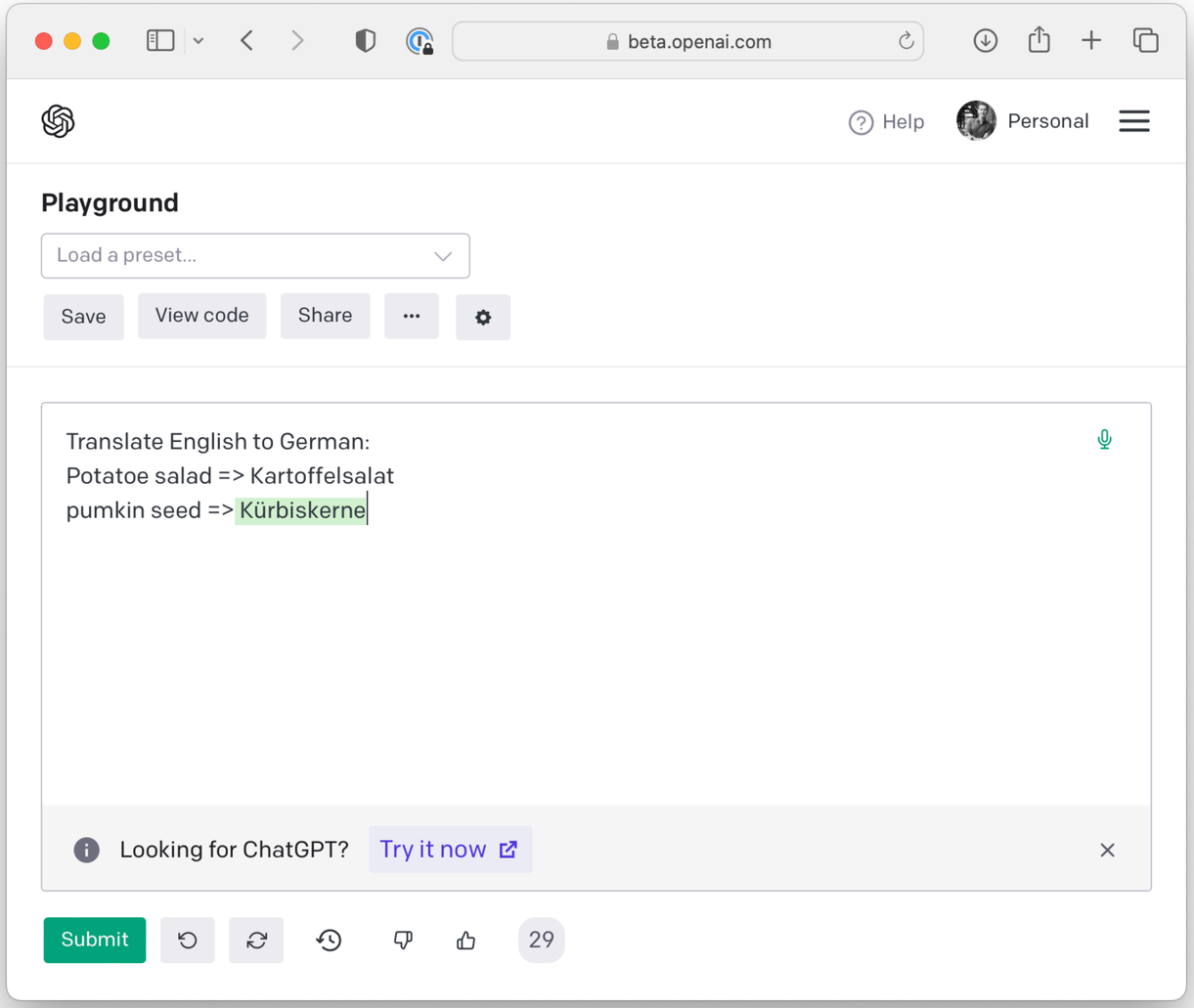


GPT 2 and 3 focused on zero- and few-shot learning via in-context learning

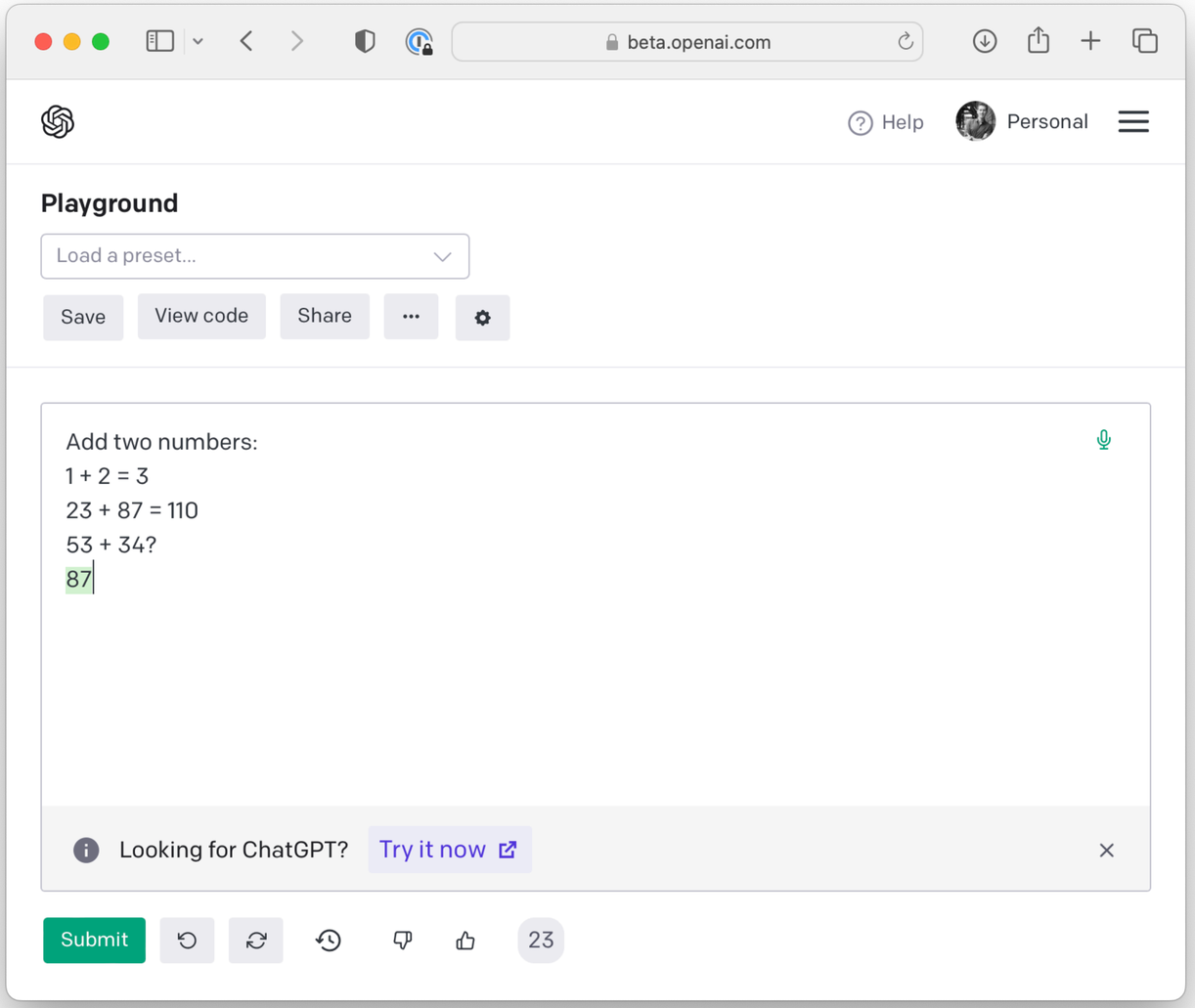
Zero-shot


beta.openai.com
Help
Personal
Playground
Load a preset...
Save
View code
Share
Translate English to German:
Potatoe Salad =>
Kartoffelsalat
Looking for ChatGPT?
Try it now
Submit
12

One-shot


beta.openai.com
Help
Personal
Playground
Load a preset...
Save
View code
Share
Translate English to German:
Potatoe salad => Kartoffelsalat
pumkin seed => Kürbiskerne
Looking for ChatGPT?
Try it now
Submit
29

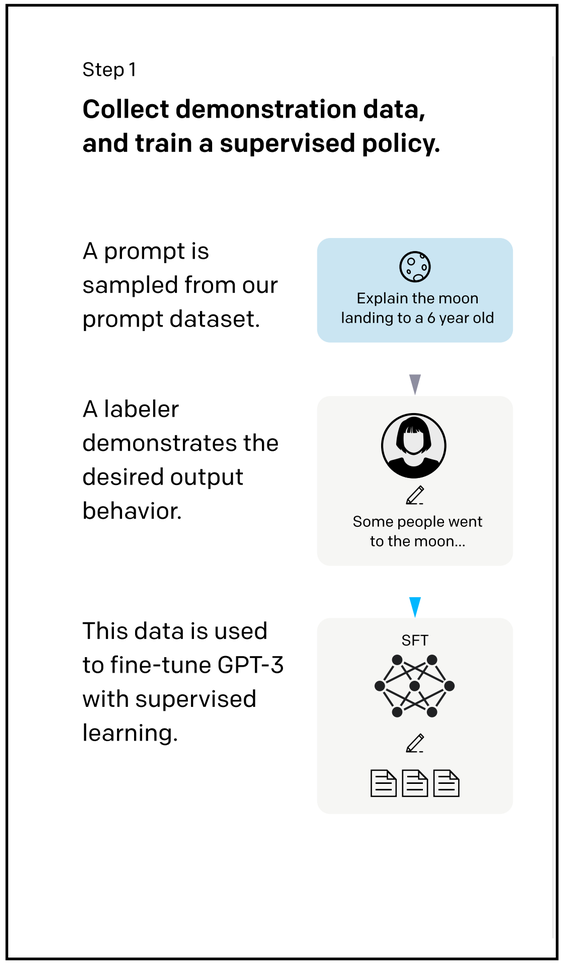
Few-shot


beta.openai.com
Help
Personal
Playground
Load a preset...
Save
View code
Share
Add two numbers:
1 + 2 = 3
23 + 87 = 110
53 + 34?
87
Looking for ChatGPT?
Try it now
Submit
23

Train a 20-billion parameter GPT model for text prediction on 3 GPU nodes with Lightning.

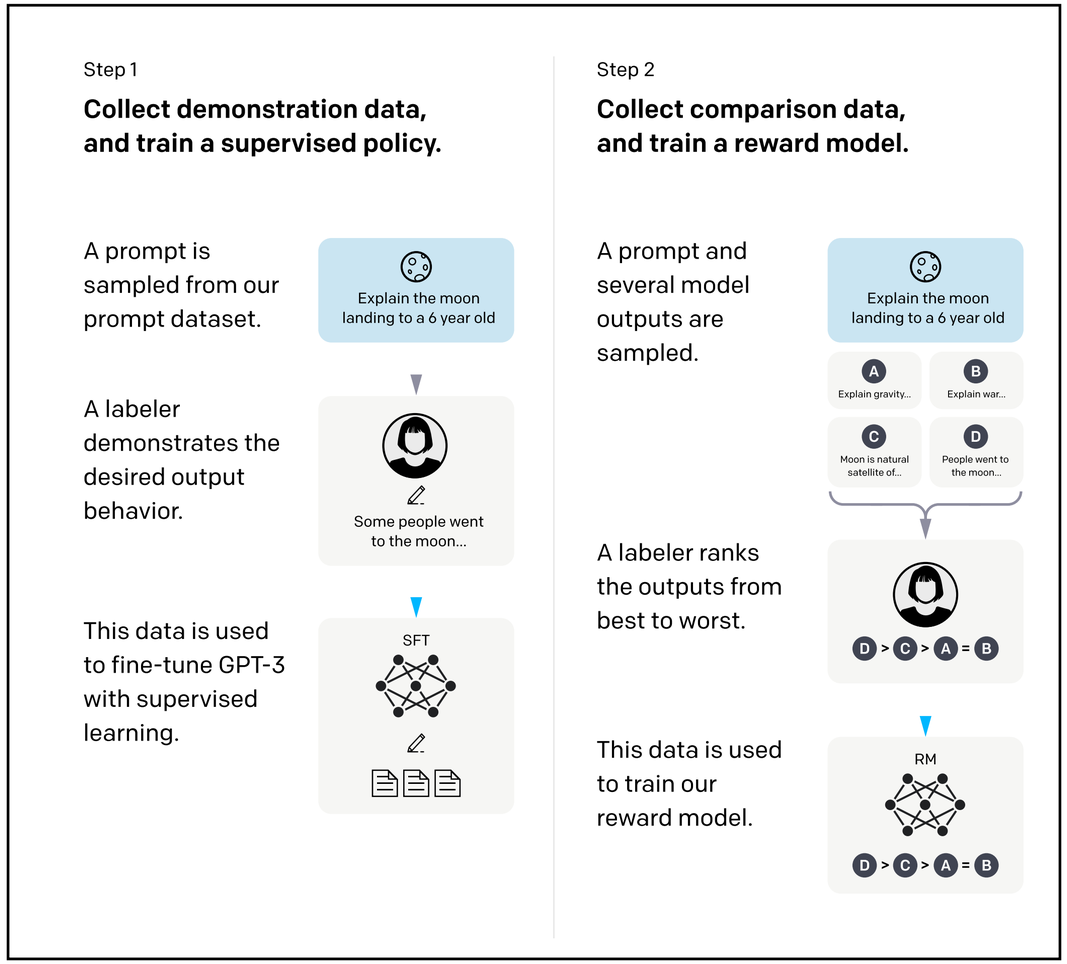
```
#! pip install light-the-torch
#! ltt install --upgrade git+https://github.com/Lightning-AI/lightning-LLMs
git+https://github.com/Lightning-AI/LAI-Text-Prediction-Component
#! curl https://cs.stanford.edu/people/karpathy/char-rnn/shakespeare_input.txt --create-dirs
-o ${HOME}/data/shakespeare/input.txt -C -


import lightning as L
import os, torch
from lightning_gpt import models
from lit_llms.tensorboard import (
    DriveTensorBoardLogger,
    MultiNodeLightningTrainerWithTensorboard,
)

from lai_textpred import default_callbacks, gpt_20b, WordDataset, error_if_local

class WordPrediction(L.LightningWork):
    def __init__(self, *args, tb_drive, **kwargs):
        super().__init__(*args, **kwargs)
        self.tensorboard_drive = tb_drive

    def run(self):
        error_if_local()

        # -------------------
        # CONFIGURE YOUR DATA
        # -------------------
        with open(os.path.expanduser("~/data/shakespeare/input.txt")) as f:
            text = f.read()
        train_dataset = WordDataset(text, 5)
        train_loader = torch.utils.data.DataLoader(
            train_dataset, batch_size=1, num_workers=4, shuffle=True
        )

        # --------------------
        # CONFIGURE YOUR MODE
        # --------------------
        model = models.DeepSpeedMinGPT(
            vocab_size=train_dataset.vocab_size,
            block_size=int(train_dataset.block_size),
            fused_adam=False,
            model_type=None,
            **gpt_20b,
        )

        # -----------------
        # RUN YOUR TRAINING
        # -----------------
        trainer = L.Trainer(
            max_epochs=2,
            limit_train_batches=250,
            precision=16,
            strategy="deepspeed_stage_3_offload",
            callbacks=default_callbacks(),
            log_every_n_steps=5,
            logger=DriveTensorBoardLogger(save_dir=".", drive=self.tensorboard_drive),
        )
        trainer.fit(model, train_loader)


app = L.LightningApp(
    MultiNodeLightningTrainerWithTensorboard(
        WordPrediction,
        num_nodes=3,
        cloud_compute=L.CloudCompute("gpu-fast-multi"),
    )
)
```

InstructGPT and ChatGPT are Additionally Trained on Human Feedback



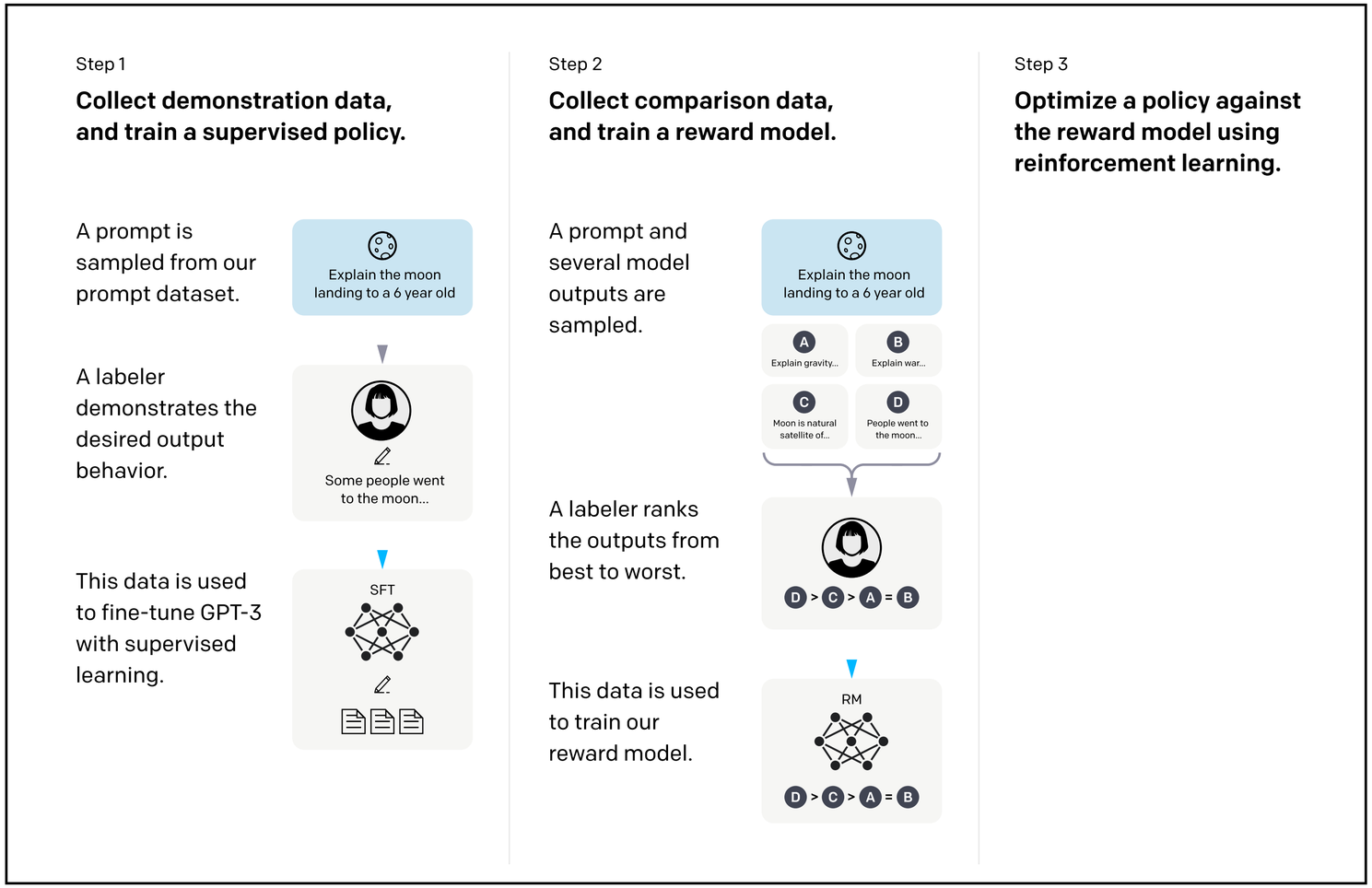
Training language models to follow instructions with human feedback, https://arxiv.org/abs/2203.02155

InstructGPT and ChatGPT are Additionally Trained on Human Feedback



Training language models to follow instructions with human feedback, https://arxiv.org/abs/2203.02155

InstructGPT and ChatGPT are Additionally Trained on Human Feedback



Training language models to follow instructions with human feedback, https://arxiv.org/abs/2203.02155

Next: A Large Language Model for Classification