

You have **2 free member-only stories left** this month. [Sign up](#) for Medium and get an extra one.

★ Member-only story

Reinforcement Learning from Human Feedback, InstructGPT, and ChatGPT



Isaac Kargar · [Follow](#)

Published in AIGuys

9 min read · Jan 7



Share

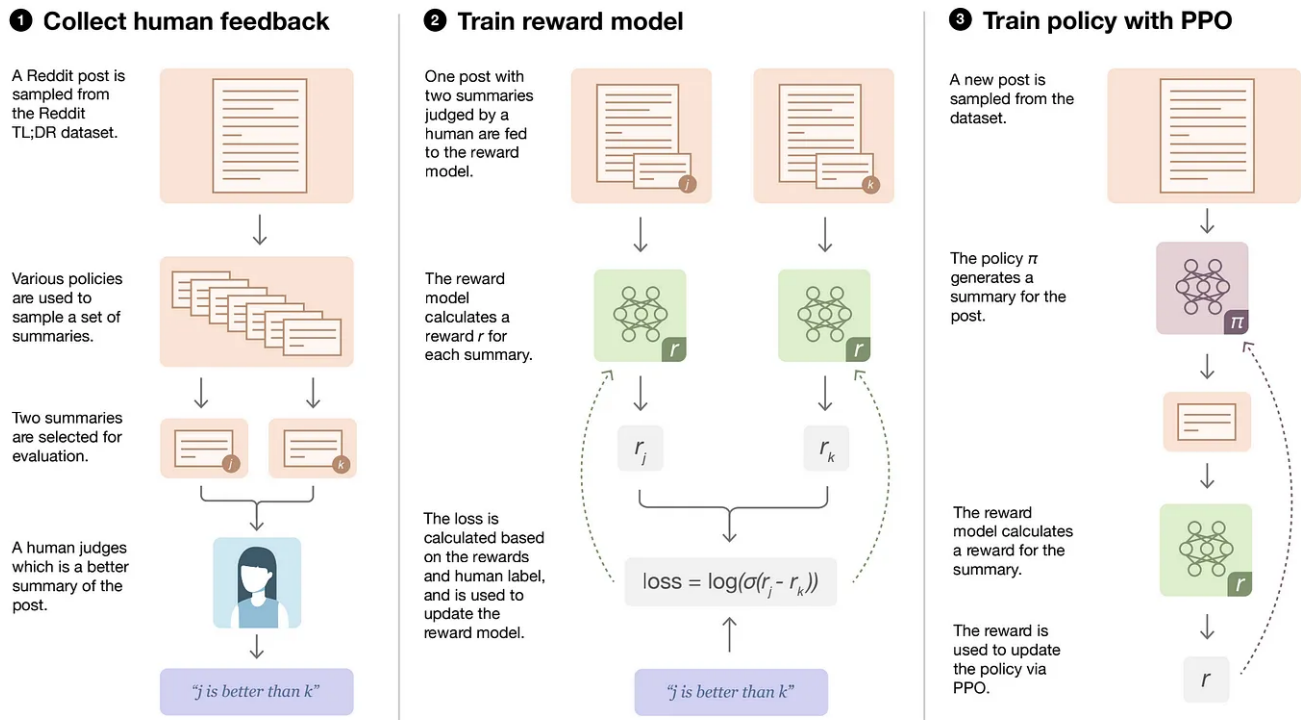
Note: some parts of this blog post are generated by ChatGPT! :)

Welcome to my blog post on ChatGPT! In this post, we will dive into the inner workings of ChatGPT and how it is trained. However, before we get into the specifics of ChatGPT, it's important to first review some relevant prior works and concepts to give us a strong foundation. Once we have a solid understanding of these foundations, we can move on to exploring ChatGPT in depth.

Let's get started.

Learning to Summarize From Human Feedback

This work demonstrates the feasibility of significantly improving summary quality through the training of a model that optimizes for human preferences. The authors collect a large dataset of human-generated comparisons between summaries, train a model to predict the summary preferred by humans, and use this model as a reward function to fine-tune a summarization policy using reinforcement learning. They showed that training with human feedback significantly outperforms strong baselines in English summarization, and also human feedback models have better generalization to new domains than supervised models.



[source](#)

They use a Reddit posts dataset and propose three steps as follows in the paper:

- For a Reddit post from the dataset, they sample summaries from several sources including the current policy, initial policy, original reference summaries, and various baselines. Humans are asked to choose the best summary for a given Reddit post from a batch of pairs of summaries. The labeler needs to provide feedback for a pair of summaries j, k like "*j is better than k*".
- Then they train a reward model using human comparisons. Given a post and two summaries judged by a labeler, the loss function is calculated based on the predicted reward r by the model for each summary, and also the human labels. Then the reward model is updated using the calculated loss. The reward model is a pre-trained model which is fine-tuned using supervised learning, with a randomly initialized linear head that outputs a scalar value. Then they train this model to predict which summary $y \in \{y_0, y_1\}$ is better as judged by a human, given a post x . If the summary preferred by the human is y_i , the loss for reward model can be written as:

$$\text{loss}(r_\theta) = E_{(x, y_0, y_1, i) \sim D} [\log(\sigma(r_\theta(x, y_i) - r_\theta(x, y_{1-i})))]$$

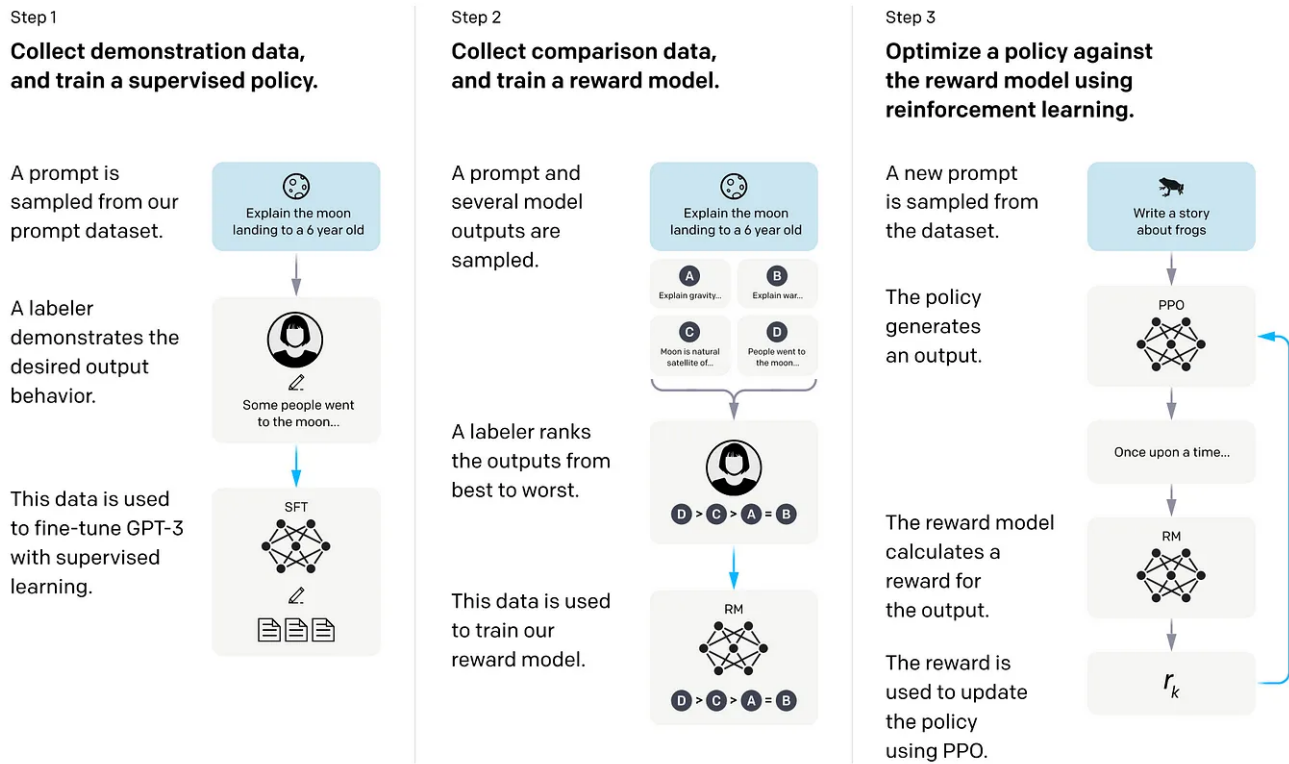
where $r_\theta(x, y)$ is the scalar output of the reward model for post x and summary y with parameters θ , and D is the dataset of human judgments. In addition, they add a KL divergence term in the reward. This KL term serves two purposes. First, it acts as an entropy bonus, encouraging the policy to explore and deterring it from collapsing to a single mode. Second, it ensures the policy doesn't learn to produce outputs that are too different from those that the reward model has seen during training.

$$R(x, y) = r_\theta(x, y) - \beta \log[\pi_\phi^{\text{RL}}(y|x) / \pi^{\text{SFT}}(y|x)]$$

- Then, they optimize the policy using the reward model as a guide. The logit output of the reward model is treated as a reward to be optimized using the PPO algorithm and reinforcement learning. The PPO policy is initialized by a model fine-tuned on the Reddit TL;DR dataset using supervised learning. For the PPO value function, they use a Transformer with completely separate parameters from the policy. They initialize the value function to the parameters of the reward model. In their experiments, the reward model, policy, and value function are the same size.

InstructGPT: Training language models to follow instructions with human feedback

This paper presents a method for aligning language models with user intent on a variety of tasks through fine-tuning with human feedback. Starting with labeler-written and API-submitted prompts, a dataset of labeler demonstrations of desired model behavior is collected and used to fine-tune GPT-3 through supervised learning. A dataset of rankings of model outputs is then collected and used to further fine-tune the supervised model with reinforcement learning and human feedback, resulting in the development of InstructGPT models. These models demonstrate improvements in truthfulness and reductions in toxic output generation while maintaining minimal performance regressions on public NLP datasets.



[source](#)

To create the initial InstructGPT models, labelers were asked to write prompts themselves. This was necessary because instruction-like prompts were not frequently submitted to regular GPT-3 models on the API, and were needed to begin the process. Three types of prompts were requested: **plain prompts where labelers were asked to come up with an arbitrary task with sufficient diversity, few-shot prompts consisting of an instruction and multiple query/response pairs**, and user-based prompts based on use cases from API waitlist applications (they had a number of use-cases stated in waitlist applications to the OpenAI API). These prompts were used to produce three datasets for fine-tuning: one with labeler demonstrations for training SFT (Supervised fine-tuning) models, one with labeler rankings of model outputs for training RMs (Reward Models), and one without human labels for RLHF (Reinforcement Learning from Human Feedback) fine-

Open in app ↗

Sign up

Sign In



extractions, and other natural language tasks.

Here is a screenshot of the web interface they used for human labels:

Submit

Skip

« Page 3 / 11 »

Total time: 05:39

Instruction
Summarize the following news article:

====
{article}
=====

Include output

Output A
summary1

Rating (1 = worst, 7 = best)

1 2 3 4 5 6 7

Fails to follow the correct instruction / task ? ☐ Yes ☐ No
Inappropriate for customer assistant ? ☐ Yes ☐ No
Contains sexual content ☐ Yes ☐ No
Contains violent content ☐ Yes ☐ No
Encourages or fails to discourage violence/abuse/terrorism/self-harm ☐ Yes ☐ No
Denigrates a protected class ☐ Yes ☐ No
Gives harmful advice ? ☐ Yes ☐ No
Expresses moral judgment ☐ Yes ☐ No

Notes

(Optional) notes

[source](#)

Here are the steps to train InstructGPT.

- First, the authors collect demonstration data and use it to train a supervised policy. The demonstration data consists of desired behavior on a specific input prompt distribution and is provided by labelers. A pre-trained GPT-3 model is then fine-tuned on this data using supervised learning to have the SFT model.
- The authors also collect comparison data, in which labelers indicate their preferred output for a given input. This data is used to train a reward model that predicts the output preferred by humans. Starting from the SFT model with the final unembedding layer removed, they trained a model to take in a prompt and response, and output a scalar reward. In this paper, the authors present labelers with a range of κ responses (from 4 to 9) to rank, instead of just presenting a pair of summaries to compare as in previous research. This results in $c(\kappa, 2)$ comparisons for each prompt shown to a labeler. To prevent overfitting and improve efficiency, the authors chose to train on all $c(\kappa, 2)$ comparisons from each prompt as a single batch element, rather than shuffling the comparisons into a single dataset and training on them one at a time. This approach is more computationally efficient because it only requires a single forward pass of the reward model for each completion, rather than $c(\kappa, 2)$ forward passes. Additionally, this method achieved improved validation accuracy and log loss

compared to the previous approach. The loss function for RM training is as follows:

$$\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x, y_w, y_l) \sim D} [\log(\sigma(r_\theta(x, y_w) - r_\theta(x, y_l)))]$$

where $r_\theta(x, y)$ is the scalar output of the reward model for prompt x and completion y with parameters θ , y_w is the preferred completion out of the pair of y_w and y_l , and D is the dataset of human comparisons.

- To further optimize the supervised policy SFT, the authors use the output of the reward model as a scalar reward and fine-tune the policy to optimize this reward using the Proximal Policy Optimization (PPO) algorithm. The objective for RL training is as follows:

$$\begin{aligned} \text{objective}(\phi) = & E_{(x, y) \sim D_{\pi_\phi^{\text{RL}}}} [r_\theta(x, y) - \beta \log(\pi_\phi^{\text{RL}}(y | x) / \pi^{\text{SFT}}(y | x))] + \\ & \gamma E_{x \sim D_{\text{pretrain}}} [\log(\pi_\phi^{\text{RL}}(x))] \end{aligned}$$

You can refer to the paper for more details.

ChatGPT

ChatGPT is a variant of GPT (Generative Pre-training Transformer), which is a transformer-based language model that was trained to generate human-like text. It is fine-tuned from a model in the GPT-3.5 series and on a large dataset of internet text and can generate coherent and coherent paragraphs of text that are difficult to distinguish from text written by humans.

Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.



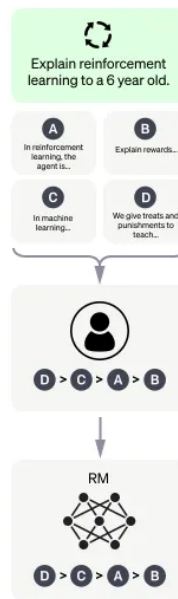
Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

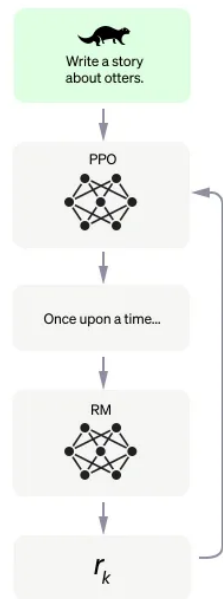
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



[source](#)

The architecture of GPT consists of an encoder and a decoder, both of which are made up of a stack of transformer blocks. The encoder processes the input text and converts it into a representation that the decoder can use to generate the output text. The decoder then generates the output text one word at a time, using the representation generated by the encoder and its own internal state to decide what the next word should be.

There is not that much new in ChatGPT compared to the previously explained papers in this post. But let's see what is happening in the different steps of ChatGPT.

Step 1 — Collect Demonstration Data and Train a Supervised Policy

A labeler samples a prompt from an available dataset and generates a desired behavior and output for that prompt. Then a pre-trained GPT-3.5 model is finetuned on the generated data, resulting in a fine-tuned, supervised model that can follow user instructions. This fine-tuned model is called the SFT (Supervised fine-tuning on human demonstrations) model.

Step 2 — Collect Comparison Data and Train a Reward Model

A user prompt is fed into the SFT model to generate several outputs for the same prompt. Then a user will assign rewards to those generated outputs and rank them which shows the quality of the responses. Then this data will be used to train a

reward model. The reward model will get the user prompt and one response and outputs the reward value proportional to the prompt.

Step 3 — Optimize a Policy Against the Reward Model Using the PPO Reinforcement Learning Algorithm

A prompt is sampled from the dataset and is passed to the supervised fine-tuned (SFT) model from step 1, which is used as the policy in a PPO reinforcement learning algorithm, to generate a response. The response is then passed through the reward model from step 2 to assess its quality, and this value is used to further fine-tune the fine-tuning model to better understand human values such as non-toxicity and factuality. This fine-tuning and policy-updating step is done using the PPO algorithm.

In conclusion, ChatGPT is an innovative and powerful language model that has the ability to generate human-like text in real-time conversation. It has the ability to understand and respond to natural language input, making it a valuable tool for a variety of applications such as customer service, language translation, and even creative writing. While it is still in the early stages of development and has limitations, ChatGPT shows great potential for improving and enhancing human-machine communication. Overall, ChatGPT is an exciting advancement in the field of natural language processing and it will be interesting to see how it continues to evolve and be utilized in the future.

Thank you for taking the time to read my post. If you found it helpful or enjoyable, please consider giving it a like and sharing it with your friends. Your support means the world to me and helps me to continue creating valuable content for you.

Resources

- [Learning to Summarize From Human Feedback](#)
- [Training language models to follow instructions with human feedback](#)

ChatGPT - Explained!



Chat GPT Rewards Model Explained!



ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for...

openai.com

Machine Learning

Deep Learning

Naturallanguageprocessing

NLP

Data Science



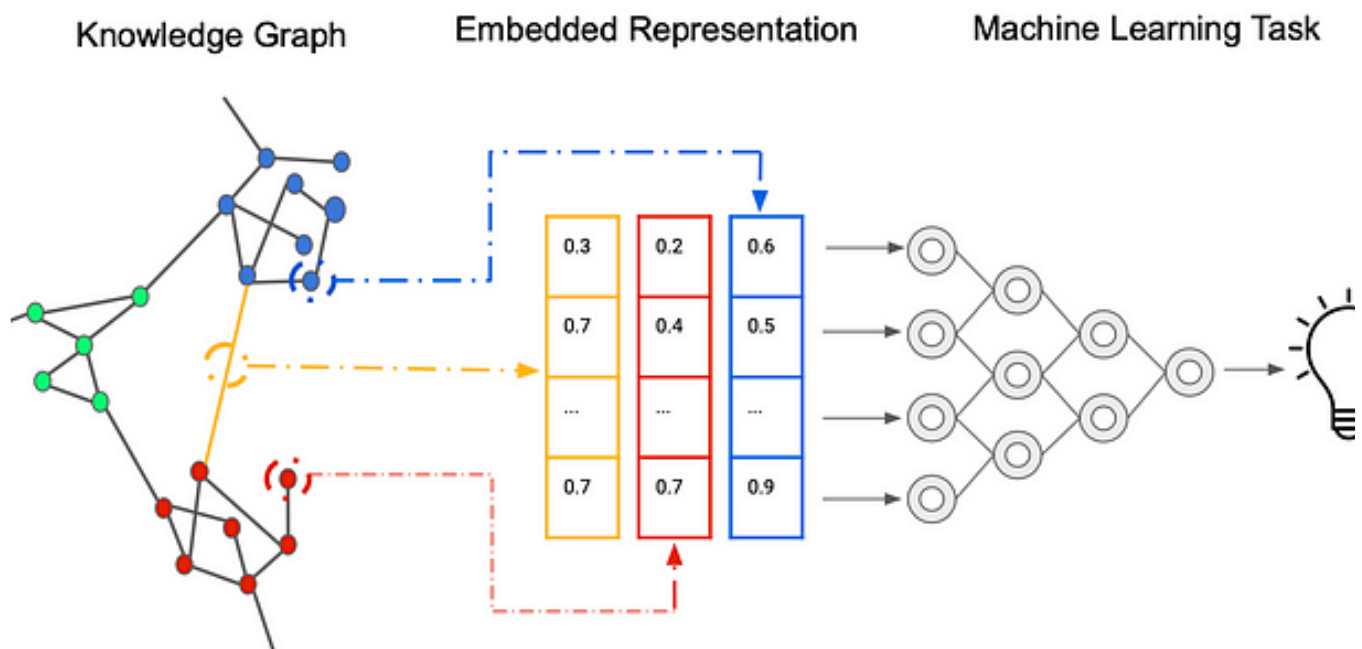
Follow

Written by Isaac Kargar

1K Followers · Writer for AIGuys

Co-Founder and CIO @ SUPPLYZ: <https://www.supplyz.eu> | Ph.D. candidate at the Intelligent Robotics Group at Aalto University | <https://kargarisaac.github.io/>

More from Isaac Kargar and AIGuys



Isaac Kargar

Graphs to Graph Neural Networks: From Fundamentals to Applications— Part 2b: Knowledge Graphs

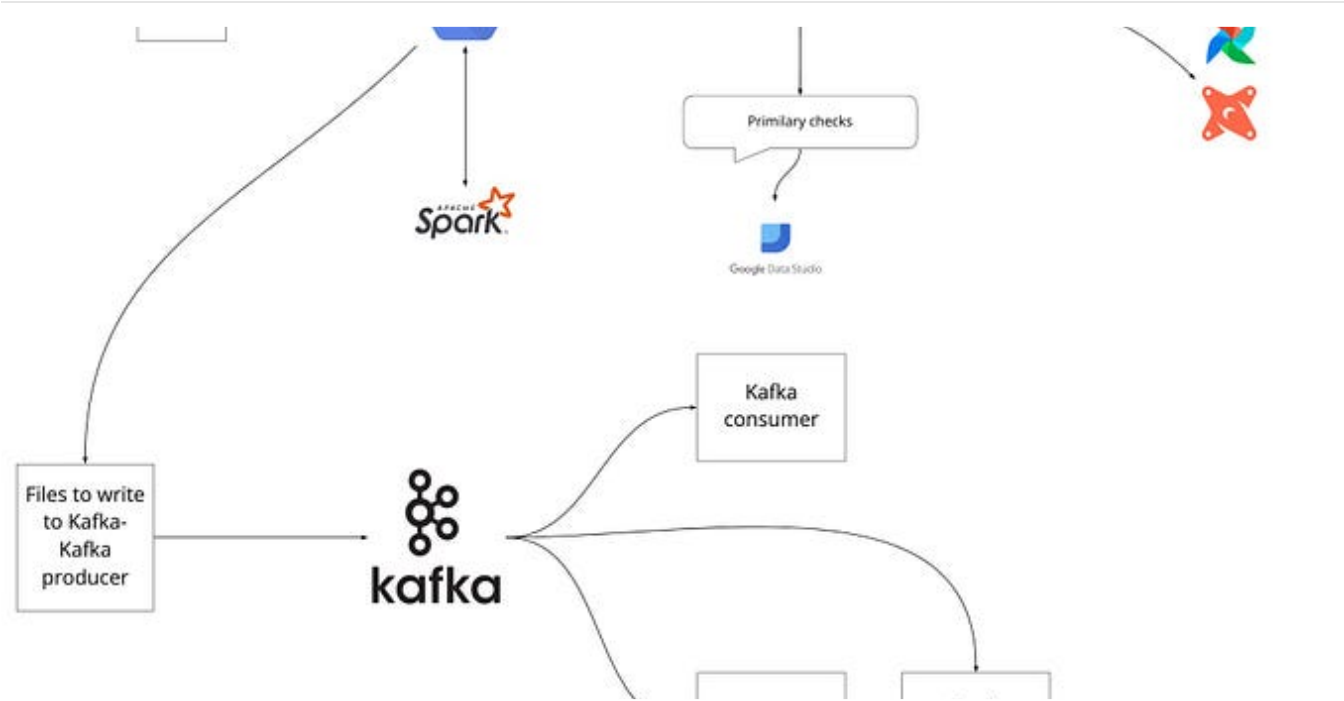
Isaac: In this post, I will continue learning about knowledge graphs. You can find the first part of this here.

🌟 · 19 min read · May 14

 14







 Isaac Kargar in AIGuys

Data Engineering—Week 1

Week 1—Data Engineering Zoomcamp course: Introduction & Prerequisites

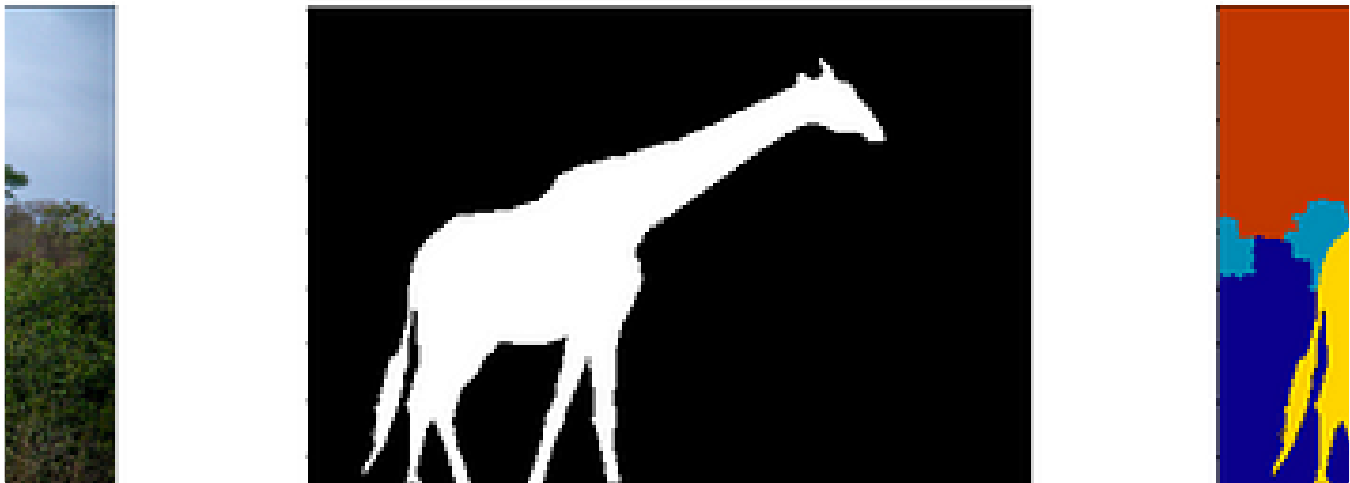
🌟 · 15 min read · Dec 28, 2022

 216

 2



Binary Mask



 Vishal Rajput in AIGuys

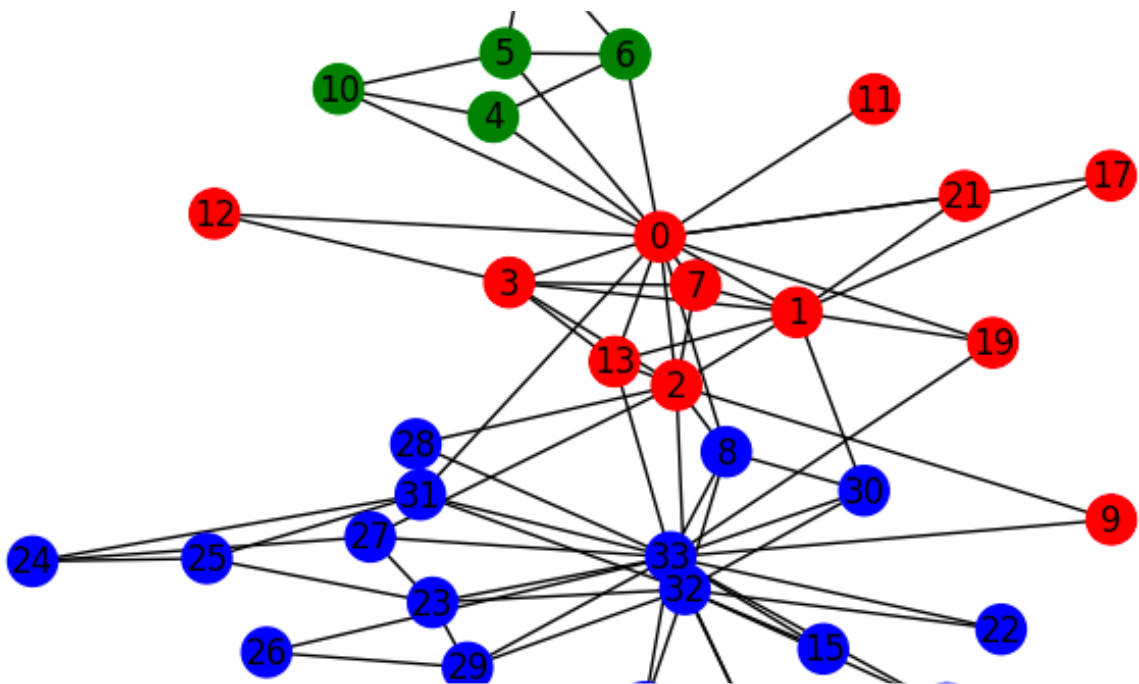
Attention U-Net, ResUnet, & many more

U-Net and all its variants

🌟 · 8 min read · Apr 1, 2022

 172 





 Isaac Kargar in AIGuys

Graphs to Graph Neural Networks: From Fundamentals to Applications— Part 1b: Graph Theory...

In this post, which is the second post from my blog post series on Graphs, we will go over another 10 questions about graph theory...

★ · 20 min read · Feb 26

👏 37 💬 2



See all from Isaac Kargar

See all from AIGuys

Recommended from Medium



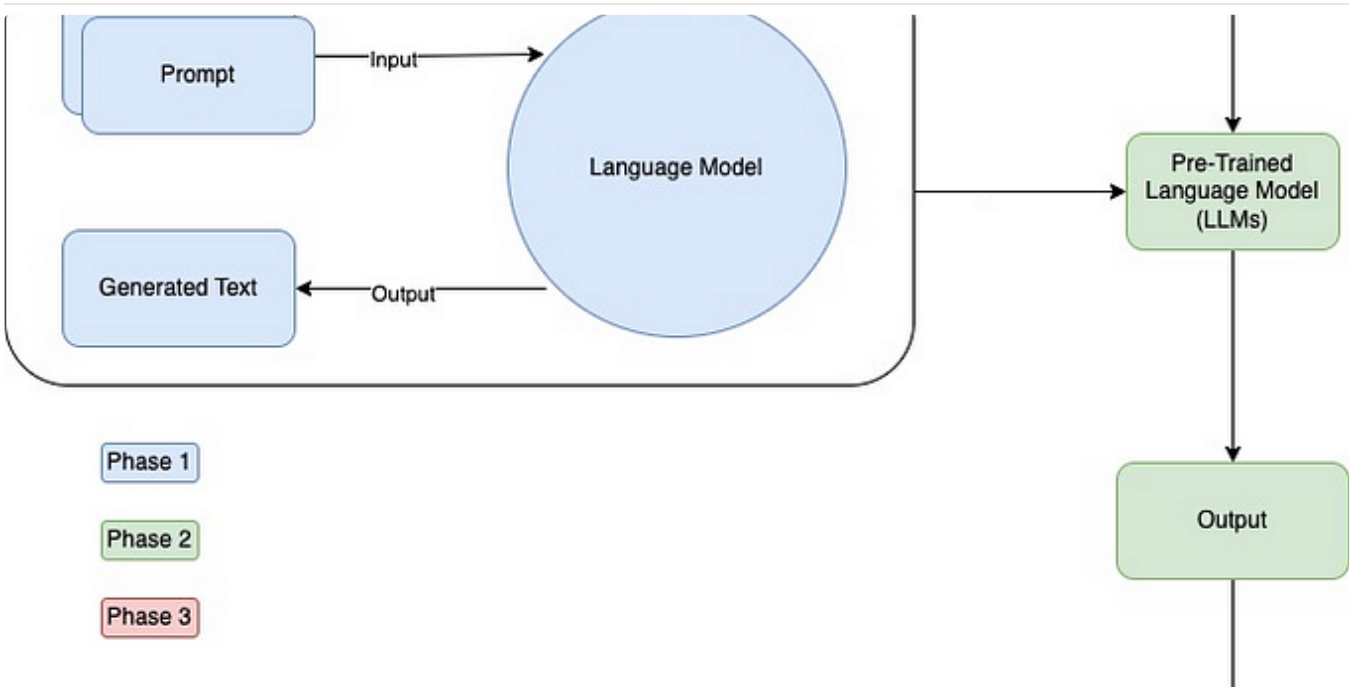
J. Qarafi

Fine-tuning the ChatGPT model

Fine-tuning the ChatGPT model is a crucial step in using the model for analyzing data related to cryptocurrencies. This step involves...

★ · 4 min read · Feb 22

52



Krishna Avva in GoPenAI

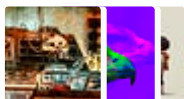
Reinforcement Learning from Human Feedback (RLHF)


Reinforcement learning with human feedback (RLHF) is a technique for training large language models (LLMs). Instead of training LLMs merely...


🌟 · 3 min read · Jan 20

41

Lists

- 

What is ChatGPT?
9 stories · 99 saves
- 

Staff Picks
348 stories · 111 saves
- 

Stories to Help You Level-Up at Work
19 stories · 96 saves



Dr. Mandar Karhade, MD. PhD. in MLearning.ai

Train Domain-Specific Model Using a Large Language Model

Keep the language and add your knowledge

★ · 5 min read · Dec 17, 2022



214



1



Wouter van Heeswijk, PhD in Towards Data Science

Proximal Policy Optimization (PPO) Explained

The journey from REINFORCE to the go-to algorithm in continuous control

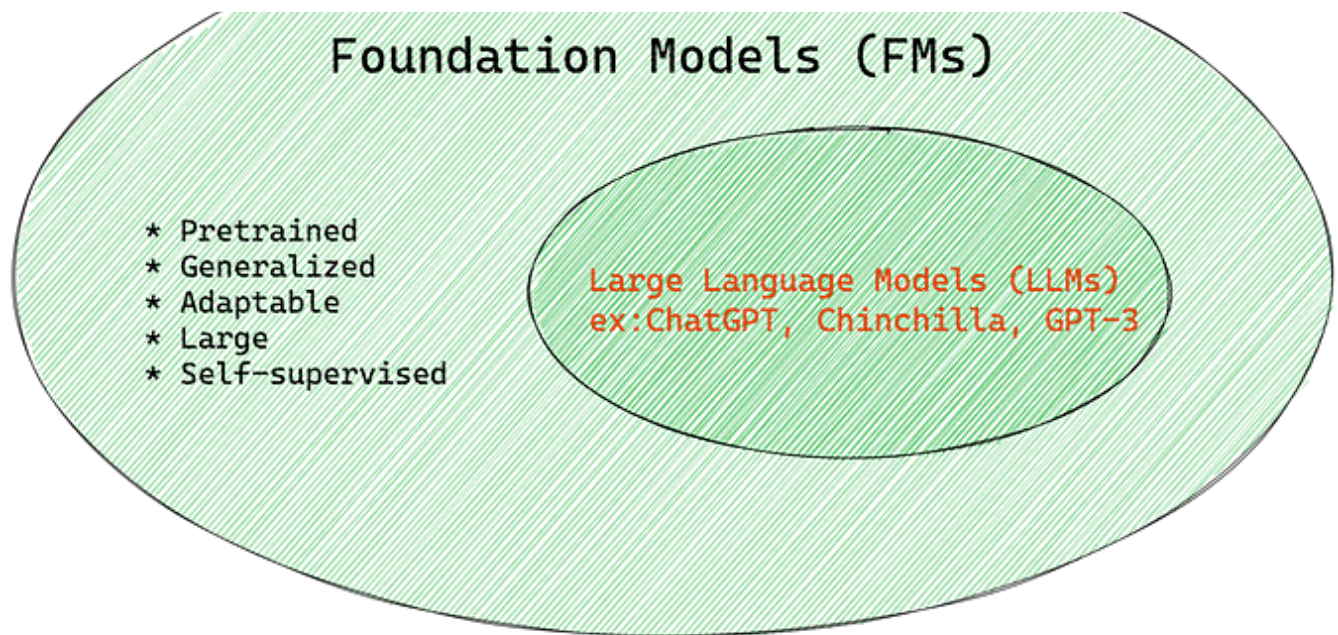
🌟 · 13 min read · Nov 30, 2022



170



2



Babar M Bhatti

Essential Guide to Foundation Models and Large Language Models

The term Foundation Model (FM) was coined by Stanford researchers to introduce a new category of ML models. They defined FMs as models...

🌟 · 15 min read · Feb 6



216





Christophe Atten in DataDrivenInvestor

Fine-tuning GPT-3 for Helpdesk Automation: A Step-by-Step Guide

Learn how to train GPT-3 on your internal database of helpdesk requests and answers using Python and the OpenAI API

🌟 · 7 min read · Jan 21



102



4



See more recommendations