Step 3 of ChatGPT Training Demystified: Part 3 of ChatGPT Series of my notes.

# Introduction

I have 2 published 2 blogs [https://www.linkedin.com/pulse/chatgpt-how-works-my-notes-part-1-ajay-taneja/ | <https://www.linkedin.com/pulse/step-2-chatgpt-training-demystified-part-series-my-blogs-ajay-taneja/>] so far on ChatGPT training. In the blog 1, I have tried and qualitatively explained the 3 fundamental steps involved in ChatGPT training as illustrated in the picture below. In the second blog, I went into finer details of step 2 of Chat GP training.

This blog will go into the finer details of Step 3 of ChatGPT training and describe the components of the Step 3. The following paragraphs will throw some light on ow Reinforcement Learning is used in ChatGPT training. Before we go into these intricacies, let us revisit the fundamental steps of ChatGPT training.

Graphical user interface

Description automatically generated with medium confidence

**Figure: Steps involved in ChatGPT Training**

# Overview of the steps of ChatGPT training

ChatGPT training can be split into 3 fundamental steps:

1. ChatGPT uses the Generative Pretrained Transformer model which is pre-trained to understand language. This pre-trained transformer model is further fine-tuned – in order to fine tune this model, we have human labellers who provide the initial prompt as well as the response. This is then fed to the network for the fine tuning. The network is trained to get the set of parameters for the fine-tuned model.
2. In the Step 2 of ChatGPT Training, we take a prompt and pass it through the same model multiple times, so that, we get “n” different responses. I have explained in the second blog[[https://www.linkedin.com/pulse/step-2-chatgpt-training-demystified-part-series-my-blogs-ajay-taneja/]](https://www.linkedin.com/pulse/step-2-chatgpt-training-demystified-part-series-my-blogs-ajay-taneja/%5d%20) about how the same language model can provide different responses to the same prompt. We then have human labellers rank these responses and the rank represents the quantification of the quality of the responses i.e., how well the “human labellers” likes the responses. The ranking happens in a scaled between 1 and 7 – the scale being termed as “Likert Scale” which is a rating system used in questionnaires that is designed to measure people’s attitudes, opinions or perceptions [<https://www.britannica.com/topic/Likert-Scale>].

The ranking used to train another “Rewards” model which takes an input prompt and response, and outputs is going to be a scalar – a “reward” – which is number between 1 and 7.

1. In the step 3 of the ChatGPT training, we’re going to use the “Rewards” model – what essentially happens in step 3 is that we pass an unseen prompt and the response to the supervised fine-tuned model, and we get a response. The response through the “Rewards” model – the Rewards model outputs a rank which is used to fine tune the parameters of the supervised fine-tuned model to output a more factual, admissible response. The resultant model then becomes the ChatGPT.

We will now demystify the Step 3 of the ChatGPT Training. In order to do that, the remaining of this blog is comprised of the following sections:

* In the section 3, we go into a bit detail of how ChatGPT generates response.
* In the section 4, we talk about the “Rewards” model being used to further fine tune the Supervised fine-tuned model of Step 1. Most importantly, in the section 4 we talk about the concept of Proximal Policy Optimisation and the corresponding loss function used that is “maximised” during the training process to generate more factual and admissible response to the prompt.

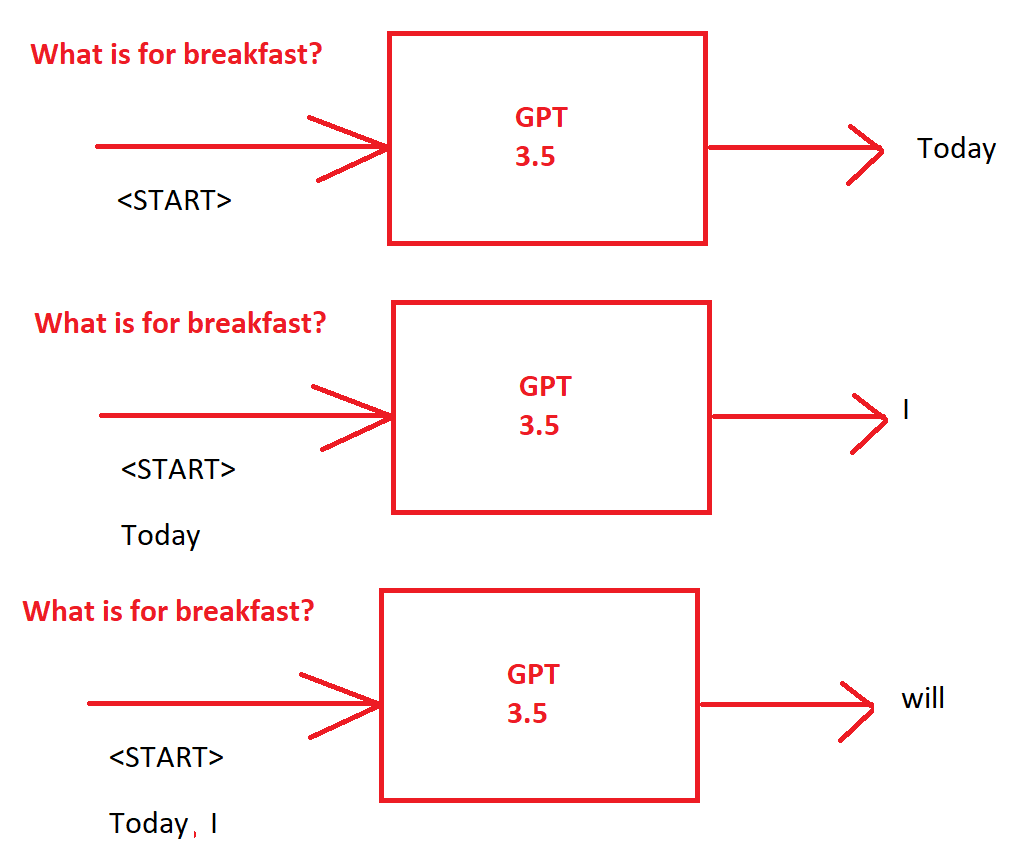
# How does ChatGPT generate response?

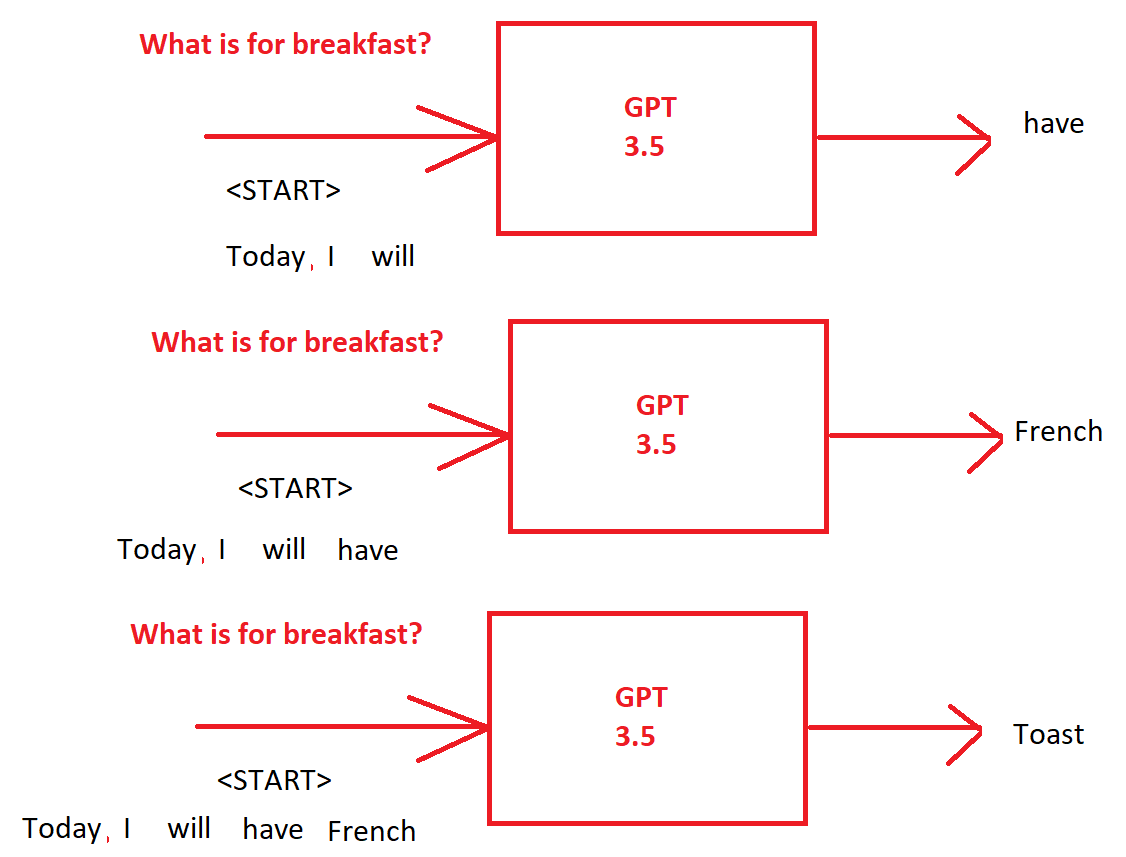
Let us now focus on the step 3 specifically and let us attempt and describe each of the components shown below in greater detail. This section will highlight how Reinforcement Learning is used in ChatGPT.

Diagram

Description automatically generated

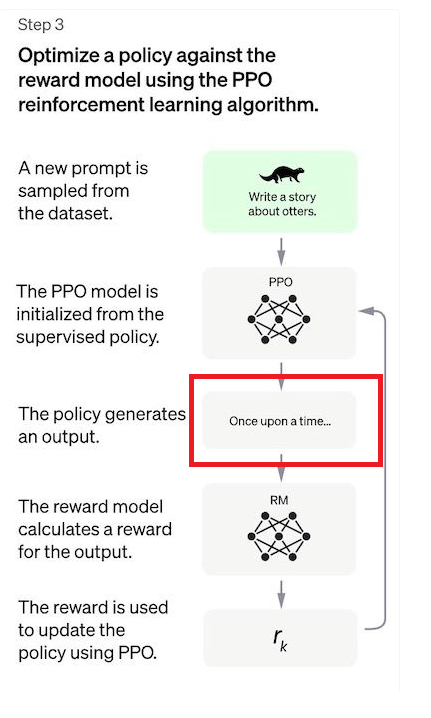
We have a Supervised Fine-tuned model to which we pass a prompt, and we generate a response. Let us see how a response looks like from GPT.





Thus, it should be underscored that the GPT (3.5/4) will generate one word at a time using all the words that had come before as an **input context.**

Thus, we have an input prompt, and we pass it to the supervised fine-tuned model to generate one word at a time (As highlighted in the figure below) until all the words have been generated for that response.



# How is the Rewards model of Step 2 utilised?

Next, the prompt and the response are passed to the “Rewards” model – that is already trained – to get the scalar value – the Reward which tells us how good the response was. We then use the Reward to fine tune the original supervised fine-tuned model of step 1. Thus, the parameters of the original supervised fine-tuned model have to be updated. Let us see how this happened next.

The parameters of the Supervised fine-tuned model are updated using the “Proximal Policy Optimisation” which are a class of techniques to maximise the reward by including the reward in the loss function itself.

**Mathematics of the loss function of the loss function of the Proximal Policy Optimisation (PPO):**

As mentioned, in the Proximal Policy Optimisation, the Reward is used in the loss function itself. Let is examine the mathematics of the Loss function in a bit detail.

The equation below is taken from the paper: Proximal Policy Optimization Algorithms

Text

Description automatically generated with low confidence

Reiterating, the Rewarded generated has to be utilized to further fine tune the Supervised Fine-tuned model. The above us the loss function used to make the gradient updates. In the above equation:

* ϴ is the parameter in the Supervised Fine-Tuned model which we will be updating
* t is the time step
* Every single time step we have one complete response that is one-time step.
* r is the Rewards ratio – it is the ratio of the rewards with the new parameters for the given input divided by the reward of the old parameters with the same given input

Therefore, if the rewards ratio is higher than 1 it means the training is going in the right direction.

* “A” is the Advantage Function – in Reinforcement learning, the advantage function assesses how high quality the output was with respect to the input. Thus, it is a number that is proportional to the reward

Thus, the product r.A is going to be very high if the response is very good.

* clip – we do not want to make the gradient update too fast – we will clip the upper and the lower bound of the ratio by ϵ - how large the ϵ is will determine how much we are allowing the gradient update to change.
* We then have an expectation – and this goes to the fact that we can generate for the same input prompt multiple kind of responses. So, we want to emulate this – we take the average of the values and so the entire proximal policy optimisation is not reliant on one output from ChatGPT. We make the gradient updates through “Gradient Ascend” as we’re maximizing the value.