Step 1 of ChatGPT Training Demystified: Part 4 of ChatGPT Series of my notes.

# Introduction

This is the continuation of my series of posts on ChatGPT training process and is the 4th blog in the series. I have talked about the following in the already published blogs:

1. The [blog 1](https://www.linkedin.com/pulse/chatgpt-how-works-my-notes-part-1-ajay-taneja/) provided an overview of the steps/fundamentals of ChatGPT training process.
2. The [blog 2](https://www.linkedin.com/pulse/step-2-chatgpt-training-demystified-part-series-my-blogs-ajay-taneja/%5d%20) went into the details about the Rewards model and the loss function that was used during the training of the Rewards model, the write up also went into the details of the reasoning of why ChatGPT might respond differently to the same prompt at different times.
3. The [blog 3](https://www.linkedin.com/pulse/step-3-chatgpt-training-demystified-part-series-my-notes-ajay-taneja/), then, went into the details of Proximal Policy Optimisation (PPO) and how he Rewards model was used to fine tune the Supervised Fine Tuned Transformer (SFT) model of step 1.

This blog will go into the specifics of step 1 of ChatGPT training process and will attempt to answer the following questions:

1. What is Generative Pre-training i.e., what is GPT in ChatGPT?!
2. What is Discriminative Fine Tuning?

The blog will go into the finer details of what exactly is a language model and how a language model is trained.

Now, firstly, as with my other blogs in ChatGPT series, before elaborating on the step 1 ChatGPT training, let us have an overview of the complete ChatGPT training process.

# Overview of the ChatGPT training process

Let us revisit the overall training process of ChatGPT in brief before going into he intricacies of the Step 1 of the ChatGPT training:

1. In the first step of ChatGPT training, we have a Generative pre-trained transformer model, thus, it already has an understanding of language. This pre-trained transformer model is fine-tuned so that it understands the user prompt and provides a response. In order to do that, human labellers provide the prompt as well as the response. Using this labelled data, the GPT model is fine tuned to geta supervised fine-tuned transformer model (SFT).
2. In the step 2 of ChatGPT training, we pass an unseen prompt to the supervised fine-tuned transformer model (SFT) and generate multiple responses from the supervised fine-tuned model I have explained in my second blog [here](https://www.linkedin.com/pulse/step-2-chatgpt-training-demystified-part-series-my-blogs-ajay-taneja/%5d%20) on why one might get different responses for the same prompt. Then, we have human labellers rank these responses based on “Likert Scale” – the ranking being based on the quality of the responses – it is the rank along with the user prompt and the response that is sued to train the Rewards model. The Rewards model tells us how high quality the response is to a specific input prompt.
3. In the step 3, we take an unseen user prompt and pass it through the supervised fine-tuned transformer model to generate a response.

We then pass the prompt and the response to the Rewards model which was trained during the Step 2 above. The Rewards model is going to output a number (rank) which indicates how high quality was the response. We use this Rank to further fine tune the Supervised Fine-Tuned Transformer model of Step 1. We fine tune the parameters of the SFT model – doing the update of the parameters over a number of iterations results in a model that is factual, non-toxic – this model becomes the ChatGPT. The steps 1 through 3 are illustrated in the picture below:

Graphical user interface, application

Description automatically generated

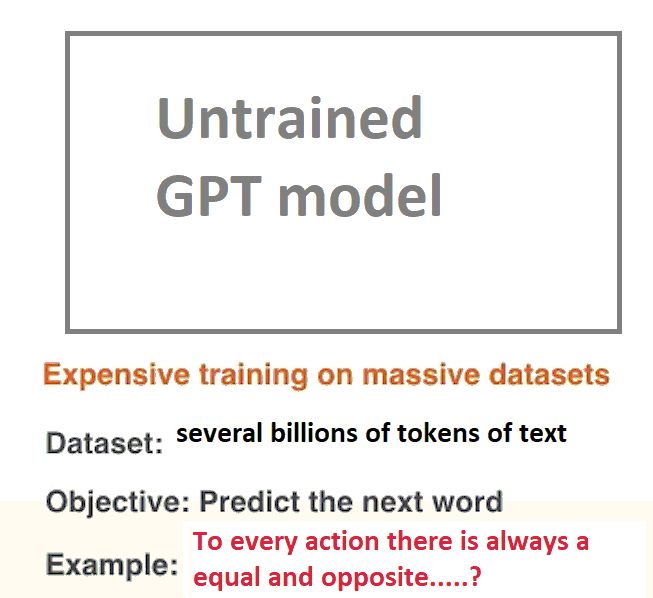
**Figure: The ChatGPT training process**

Let us now go into the intricacies of how we construct the GPT model to begin with, what is GPT and what is Discriminative Fine Tuning

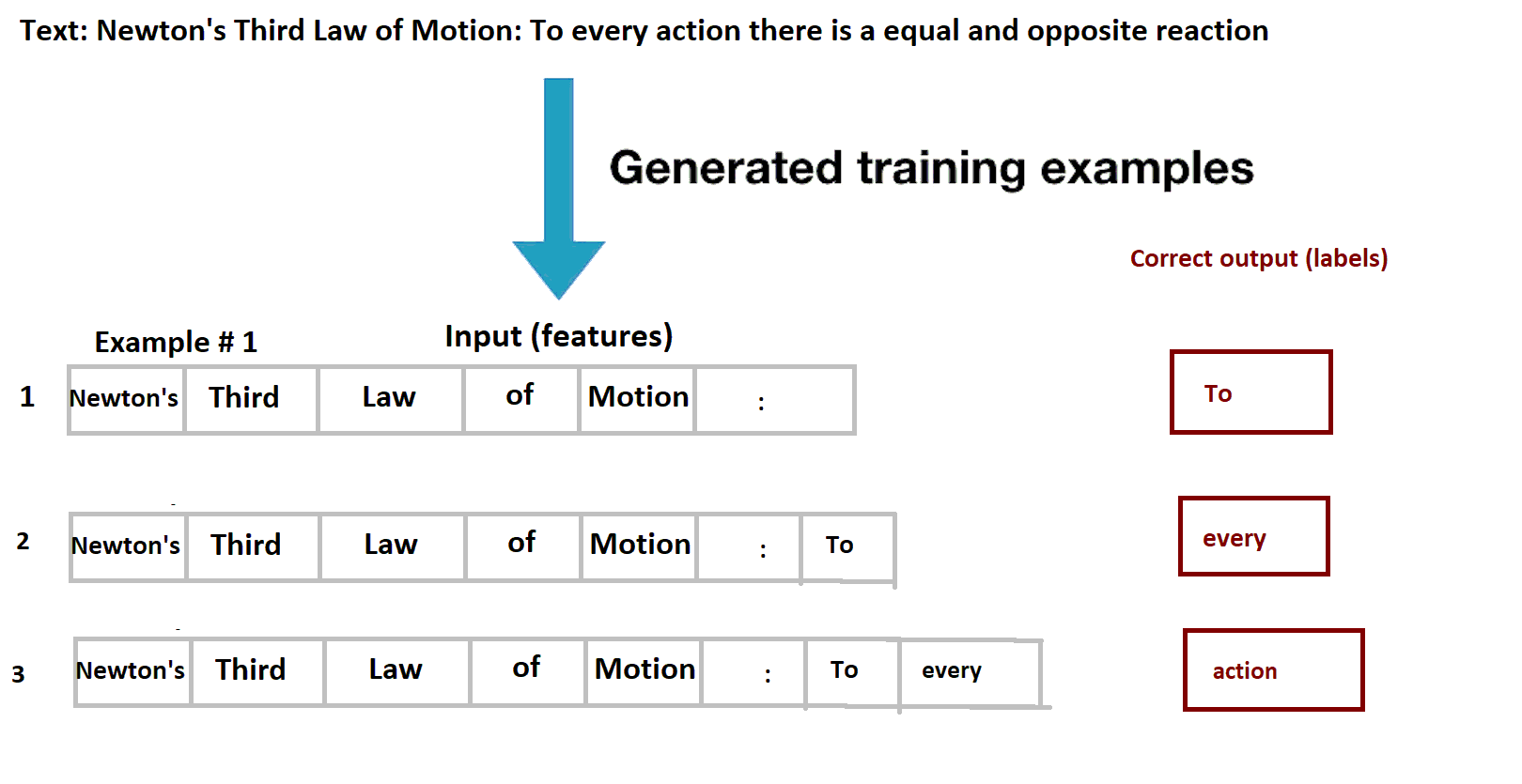
# Generative Pre-trained model: What is GPT in ChatGPT?

I have mentioned in the above sections, that in the step 1 of ChatGPT training that the Pre-trained (GPT) model is fine tuned to – to that we had the labellers who provide the prompt and a response and the parameters of the GPT model are fine tuned. Let us now start see how this GPT model is trained – let us start from an “untrained” model and see how we get GPT.

In the training phase of the GPT, the model is trained with a lot of data which includes several billions of tokens. We can think of a token as a word – the model was trained on a specific task – the task being to predict the “next word”. That is, we give a sequence of words to the model, and we ask it to predict the next word. The model in this manner after several billions of steps captures something.

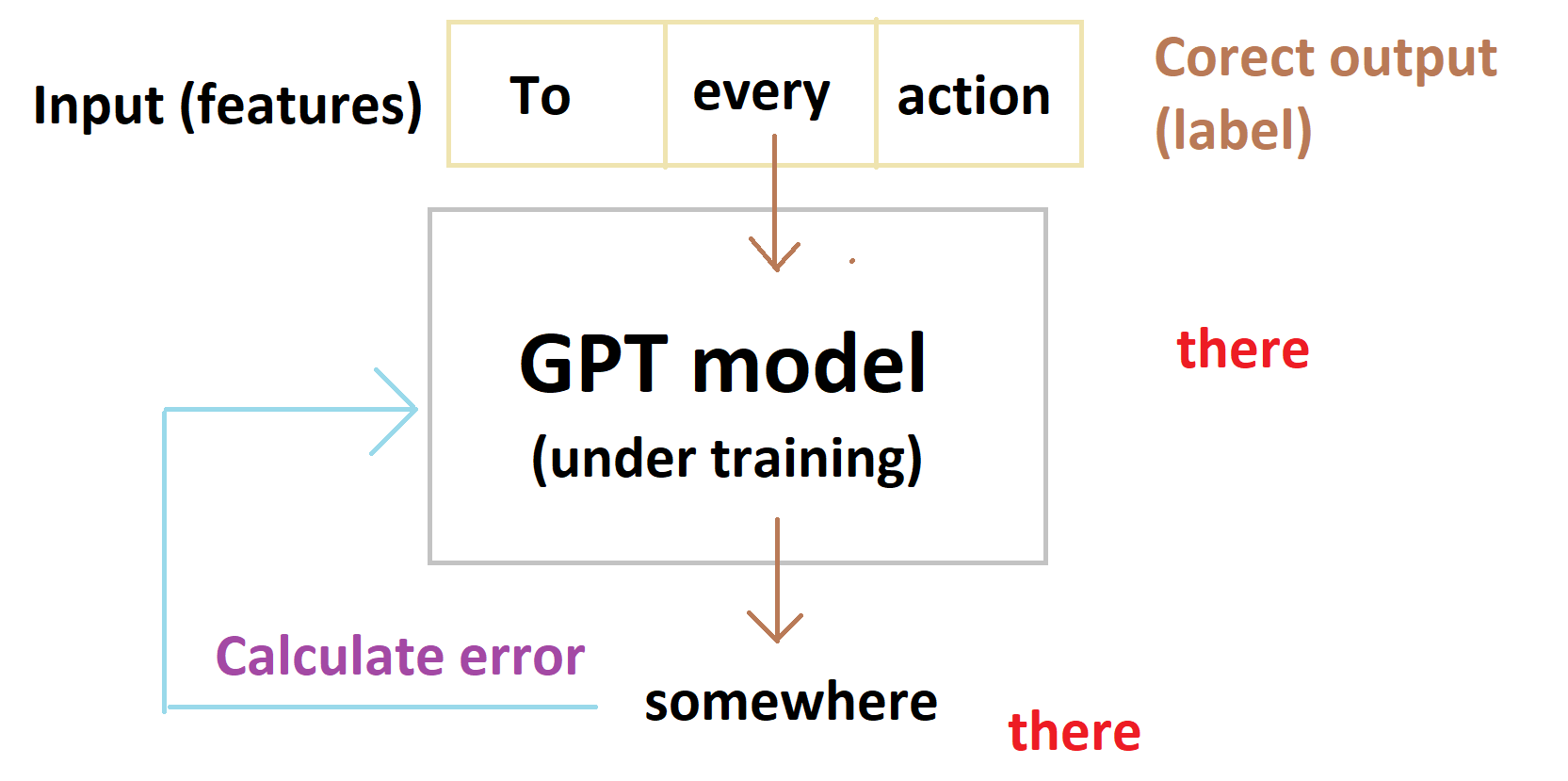


**Figure: Pre-training of the GPT model**



**Figure: Pre-training of the GPT model**

One can provide training examples to predict the next word. If you crawl the internet and get texts from newspapers, websites, test books – we end up with lots of text and we can generate billions of examples. So, we present such examples and make the model learn. So, we go through the same process as in normal training.



**Figure: Pre-training of the GPT model**

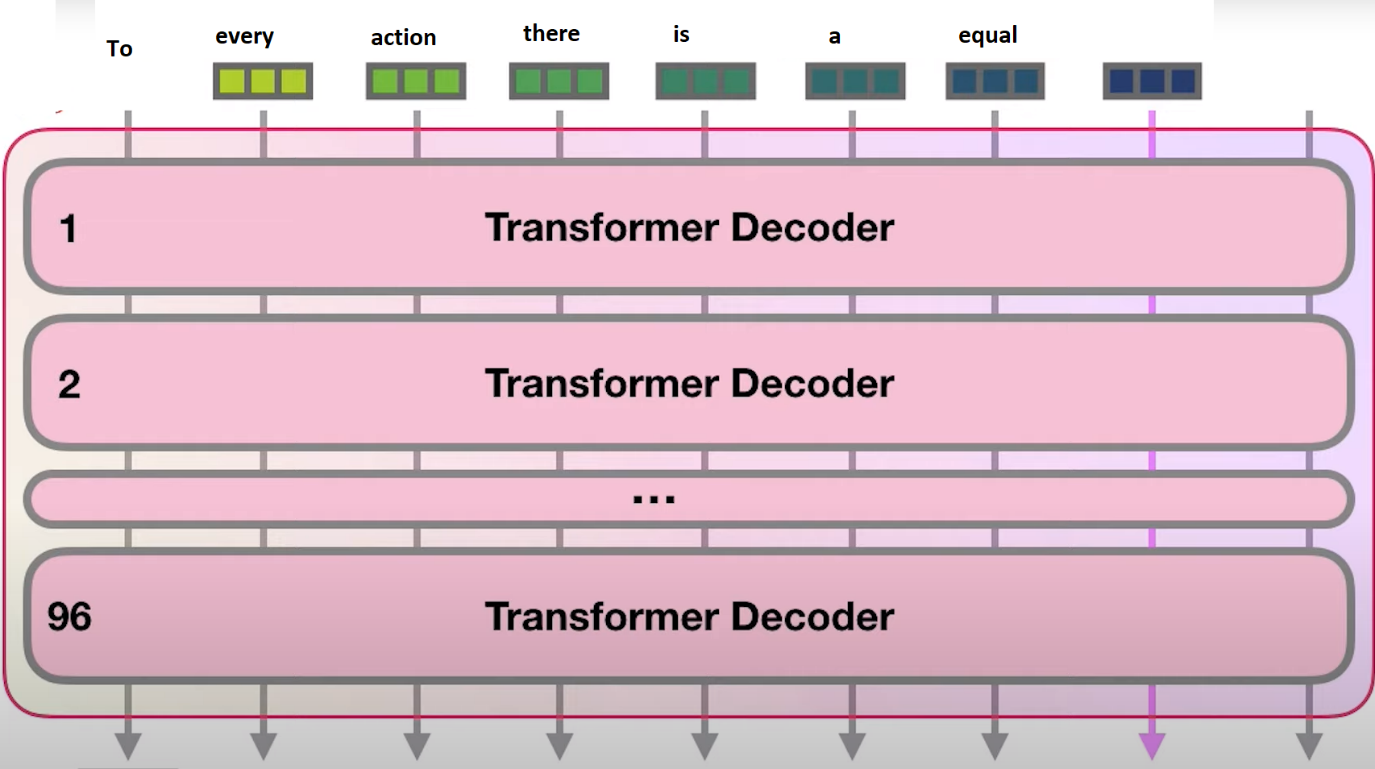
* The model generates the next word.
* We calculate the error.
* We calculate the loss function.
* We update the weight or the parameter.

This is the training process which is not new. If one has come across Machine Learning training before, this loop is not new.

**Understanding the process, a bit more:**

Understanding how the process works a bit more – you have the words, and you have the vector embedding for the word and you have the vector go through the layers of the transformer decoder. We have 96 of these – each word embedding passes through the decoder, and it gets processed. Finally, when all the tokens have processed in the fashion, we get the output.

By doing the process – we get the model understand language – its has now been “pre-trained”.



**Figure: Transformer Decoders**

# Why did we do the Pre-training and what is objective function for generative pre-training?

**Objective Function for Generative Pre-training**

As pointed out, with Generative Pre-training, we’ve trained a language model. Language models have an understanding of word sequences, and the main objective of a language model is to predict what word is going to come next given the context of all the previous words that have come before it. Mathematically, the objective function can be represented as:

Objective function:

Thus, mathematically we want the Generative Pre-trained Transformer (GPT) to optimize some objective which is typical of Machine Learning and Deep Learning models. The objective we’re optimizing is the language model objective where want to predict the next word using all the words that come before it

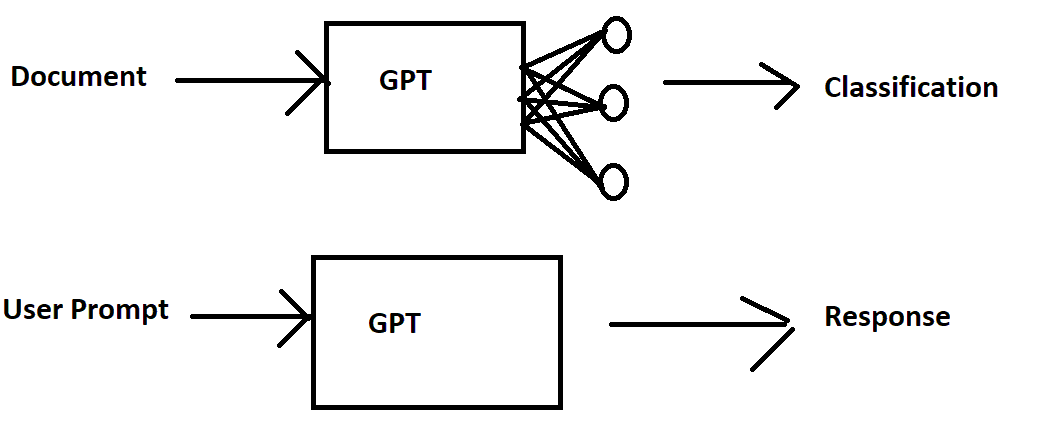
Once we’ve trained the language model as above, the model will have understanding of language – given a word sequence it will be able to figure out the next word.

**Why did we do the Pre-training?**

We did the pre-training because we wanted the model to have the notion of language which is important. Once the model has the notion of language, we can fine tune the model for specific tasks such as: Question-Answering, Document Classification and Sentiment Analysis. This will be the Discriminative Fine-Tuning phase where model is fine tuned for specific tasks.

# Discriminative Fine-Tuning Phase

Now, we have a model which has an understanding of language – in the discriminative fine-tuning phase, we want to solve a very specific problem. IT could be question-answering, sentiment analysis or document classification or any other problem associated with language. For example, in document classification, giving the next word is not enough, we want to understand the overall sentiment of the document of what it represents, or we want to understand a categorization of the document – of whether it is a fatigue analysis report, stress analysis report / likewise. Typically we have a pre-trained model, we will add a linear layer, add a small amount of training data for document classification basically learn a small set of parameters and further fine tune the model



Because there is a small set of parameters, it makes it much easier for fine tuning. For chatbot and ChatGPT, the input is a prompt and output are a response and the pre-trained model is fine tuned.

And it is this, in the first step we see wherein the labeller demonstrates the desired output behaviour and the data is used to fine tune with supervised learning.

Summarizing:

To every action there is always and equal and opposite ------🡪

This is going to be applied not a softmax because we want to convert the prediction of next word to a probability distribution – we do not choose the word with the highest probability, but we do by sampling

