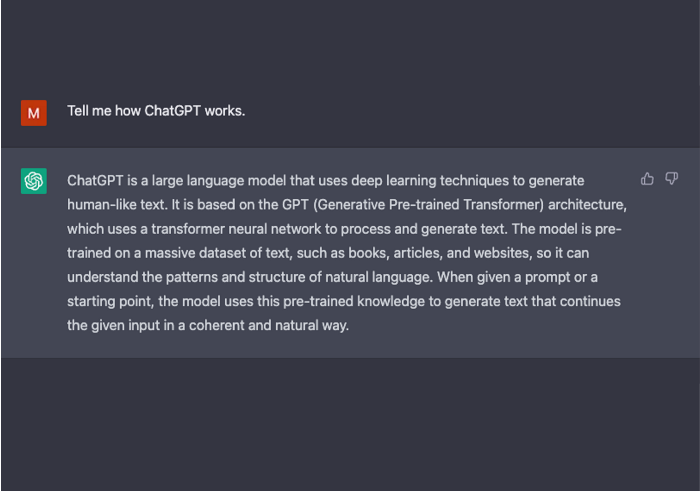
**How ChatGPT Works: The Model Behind The Bot**

A brief introduction to the intuition and methodology behind the chat bot you can’t stop hearing about.

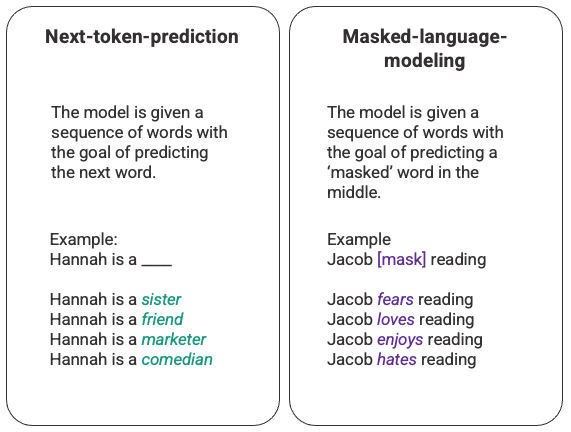


This gentle introduction to the machine learning models that power ChatGPT, will start at the introduction of Large Language Models, dive into the revolutionary self-attention mechanism that enabled GPT-3 to be trained, and then burrow into Reinforcement Learning From Human Feedback, the novel technique that made ChatGPT exceptional.

**Large Language Models**

ChatGPT is an extrapolation of a class of machine learning Natural Language Processing models known as Large Language Model (LLMs). LLMs digest huge quantities of text data and infer relationships between words within the text. These models have grown over the last few years as we’ve seen advancements in computational power. LLMs increase their capability as the size of their input datasets and parameter space increase.

The most basic training of language models involves predicting a word in a sequence of words. Most commonly, this is observed as either next-token-prediction and masked-language-modeling.



Arbitrary example of next-token-prediction and masked-language-modeling generated by the author.

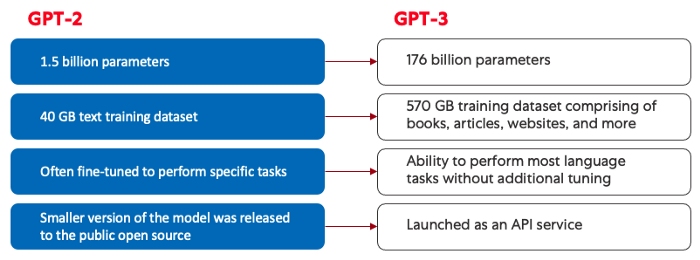
In this basic sequencing technique, often deployed through a Long-Short-Term-Memory (LSTM) model, the model is filling in the blank with the most statistically probable word given the surrounding context. There are two major limitations with this sequential modeling structure.

1. The model is unable to value some of the surrounding words more than others. In the above example, while ‘reading’ may most often associate with ‘hates’, in the database ‘Jacob’ may be such an avid reader that the model should give more weight to ‘Jacob’ than to ‘reading’ and choose ‘love’ instead of ‘hates’.
2. The input data is processed individually and sequentially rather than as a whole corpus. This means that when an LSTM is trained, the window of context is fixed, extending only beyond an individual input for several steps in the sequence. This limits the complexity of the relationships between words and the meanings that can be derived.

In response to this issue, in 2017 a team at Google Brain introduced transformers. Unlike LSTMs, transformers can process all input data simultaneously. Using a self-attention mechanism, the model can give varying weight to different parts of the input data in relation to any position of the language sequence. This feature enabled massive improvements in infusing meaning into LLMs and enables processing of significantly larger datasets.

**GPT and Self-Attention**

Generative Pre-training Transformer (GPT) models were first launched in 2018 by openAI as GPT-1. The models continued to evolve over 2019 with GPT-2, 2020 with GPT-3, and most recently in 2022 with InstructGPT and ChatGPT. Prior to integrating human feedback into the system, the greatest advancement in the GPT model evolution was driven by achievements in computational efficiency, which enabled GPT-3 to be trained on significantly more data than GPT-2, giving it a more diverse knowledge base and the capability to perform a wider range of tasks.



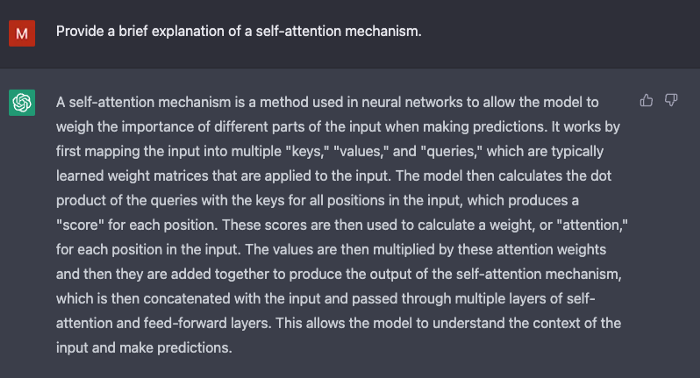
Comparison of GPT-2 (left) and GPT-3 (right). Generated by the author.

All GPT models leverage the transformer architecture, which means they have an encoder to process the input sequence and a decoder to generate the output sequence. Both the encoder and decoder have a multi-head self-attention mechanism that allows the model to differentially weight parts of the sequence to infer meaning and context. In addition, the encoder leverages masked-language-modeling to understand the relationship between words and produce more comprehensible responses.

The self-attention mechanism that drives GPT works by converting tokens (pieces of text, which can be a word, sentence, or other grouping of text) into vectors that represent the importance of the token in the input sequence. To do this, the model,

1. Creates a query, key, and value vector for each token in the input sequence.
2. Calculates the similarity between the query vector from step one and the key vector of every other token by taking the dot product of the two vectors.
3. Generates normalized weights by feeding the output of step 2 into a [softmax function](https://deepai.org/machine-learning-glossary-and-terms/softmax-layer).
4. Generates a final vector, representing the importance of the token within the sequence by multiplying the weights generated in step 3 by the value vectors of each token.

The ‘multi-head’ attention mechanism that GPT uses is an evolution of self-attention. Rather than performing steps 1–4 once, the model iterates this mechanism several times, each time generating a new linear projection of the query, key, and value vectors. By expanding self-attention in this way, the model is capable of grasping sub-meanings and more complex relationships within the input data.



Screenshot from ChatGPT generated by the author.

Although GPT-3 introduced remarkable advancements in natural language processing, it is limited in its ability to align with user intentions. For example, GPT-3 may produce outputs that

* **Lack of helpfulness**meaning they donot follow the user’s explicit instructions.
* **Contain hallucinations**that reflect non-existing or incorrect facts.
* **Lack interpretability** making it difficult for humans to understand how the model arrived at a particular decision or prediction.
* **Include toxic or biased content** that is harmful or offensive and spreads misinformation.

Innovative training methodologies were introduced in ChatGPT to counteract some of these inherent issues of standard LLMs.

**ChatGPT**

ChatGPT is a spinoff of InstructGPT, which introduced a novel approach to incorporating human feedback into the training process to better align the model outputs with user intent. Reinforcement Learning from Human Feedback (RLHF) is described in depth in [openAI’s 2022](https://arxiv.org/pdf/2203.02155.pdf) paper **Training language models to follow instructions with human feedback**and is simplified below.

**Step 1: Supervised Fine Tuning (SFT) Model**

The first development involved fine-tuning the GPT-3 model by hiring 40 contractors to create a supervised training dataset, in which the input has a known output for the model to learn from. Inputs, or prompts, were collected from actual user entries into the Open API. The labelers then wrote an appropriate response to the prompt thus creating a known output for each input. The GPT-3 model was then fine-tuned using this new, supervised dataset, to create GPT-3.5, also called the SFT model.

In order to maximize diversity in the prompts dataset, only 200 prompts could come from any given user ID and any prompts that shared long common prefixes were removed. Finally, all prompts containing personally identifiable information (PII) were removed.

After aggregating prompts from OpenAI API, labelers were also asked to create sample prompts to fill-out categories in which there was only minimal real sample data. The categories of interest included

* **Plain prompts:** any arbitrary ask.
* **Few-shot prompts:**instructions that contain multiple query/response pairs.
* **User-based prompts:**correspond to a specific use-case that was requested for the OpenAI API.

When generating responses, labelers were asked to do their best to infer what the instruction from the user was. The paper describes the main three ways that prompts request information.

1. **Direct:** “Tell me about…”
2. **Few-shot:** Given these two examples of a story, write another story about the same topic.
3. **Continuation:**Given the start of a story, finish it.

The compilation of prompts from the OpenAI API and hand-written by labelers resulted in 13,000 input / output samples to leverage for the supervised model.

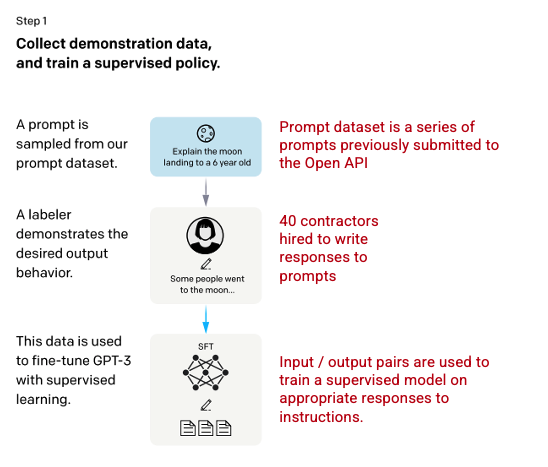
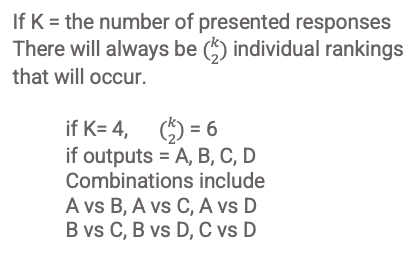


Image (left) inserted from Training language models to follow instructions with human feedback *OpenAI et al., 2022*<https://arxiv.org/pdf/2203.02155.pdf>. Additional context added in red (right) by the author.

**Step 2: Reward Model**

After the SFT model is trained in step 1, the model generates better aligned responses to user prompts. The next refinement comes in the form of training a reward model in which a model input is a series of prompts and responses, and the output is a scaler value, called a reward. The reward model is required in order to leverage Reinforcement Learning in which a model learns to produce outputs to maximize its reward (see step 3).

To train the reward model, labelers are presented with 4 to 9 SFT model outputs for a single input prompt. They are asked to rank these outputs from best to worst, creating combinations of output ranking as follows.



Example of response ranking combinations. Generated by the author.

Including each combination in the model as a separate datapoint led to overfitting (failure to extrapolate beyond seen data). To solve, the model was built leveraging each group of rankings as a single batch datapoint.

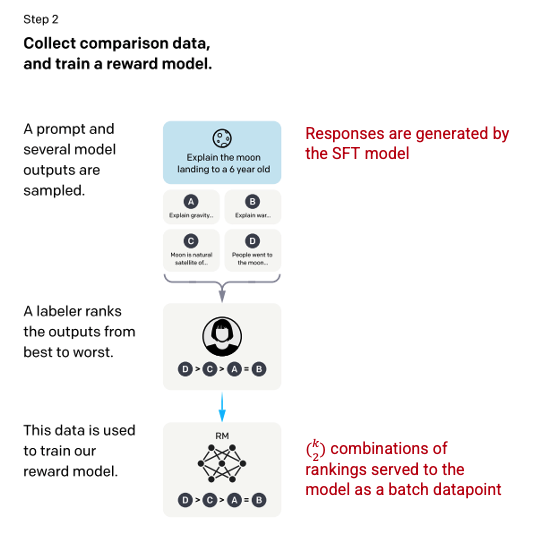


Image (left) inserted from Training language models to follow instructions with human feedback *OpenAI et al., 2022*<https://arxiv.org/pdf/2203.02155.pdf>. Additional context added in red (right) by the author.

**Step 3: Reinforcement Learning Model**

In the final stage, the model is presented with a random prompt and returns a response. The response is generated using the ‘policy’ that the model has learned in step 2. The policy represents a strategy that the machine has learned to use to achieve its goal; in this case, maximizing its reward. Based on the reward model developed in step 2, a scaler reward value is then determined for the prompt and response pair. The reward then feeds back into the model to evolve the policy.

In 2017, Schulman *et al.*introduced [Proximal Policy Optimization (PPO)](https://towardsdatascience.com/proximal-policy-optimization-ppo-explained-abed1952457b), the methodology that is used in updating the model’s policy as each response is generated. PPO incorporates a per-token Kullback–Leibler (KL) penalty from the SFT model. The KL divergence measures the similarity of two distribution functions and penalizes extreme distances. In this case, using a KL penalty reduces the distance that the responses can be from the SFT model outputs trained in step 1 to avoid over-optimizing the reward model and deviating too drastically from the human intention dataset.

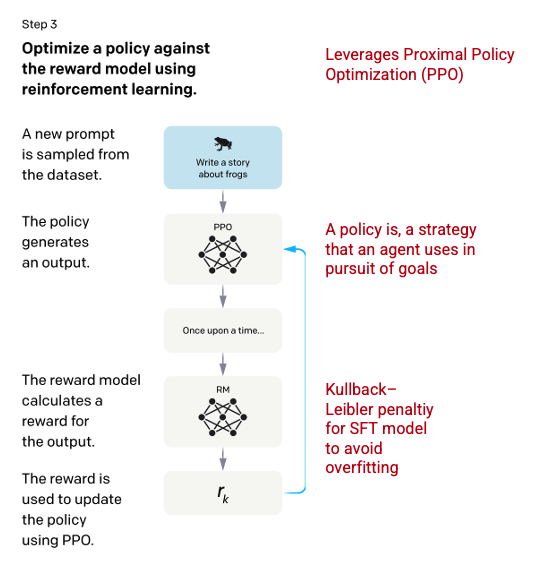
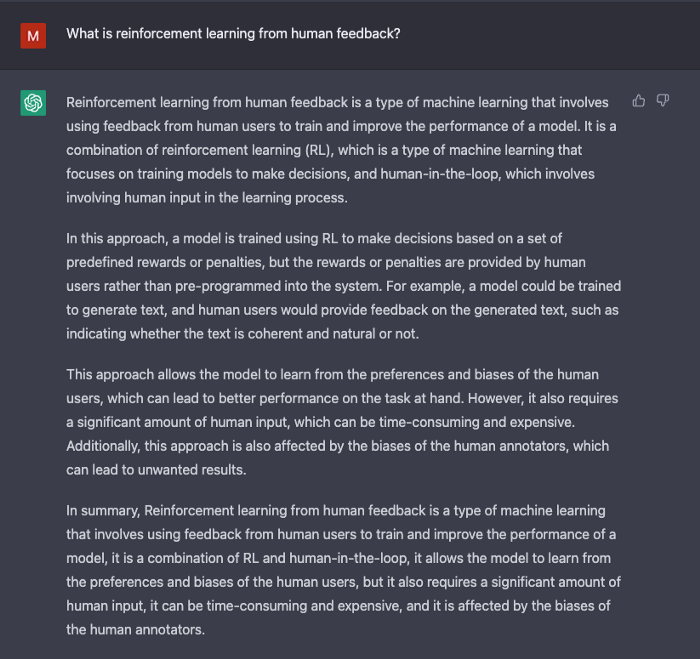


Image (left) inserted from Training language models to follow instructions with human feedback *OpenAI et al., 2022*<https://arxiv.org/pdf/2203.02155.pdf>. Additional context added in red (right) by the author.

Steps 2 and 3 of the process can be iterated through repeatedly though in practice this has not been done extensively.



Screenshot from ChatGPT generated by the author.

**Evaluation of the Model**

Evaluation of the model is performed by setting aside a test set during training that the model has not seen. On the test set, a series of evaluations are conducted to determine if the model is better aligned than its predecessor, GPT-3.

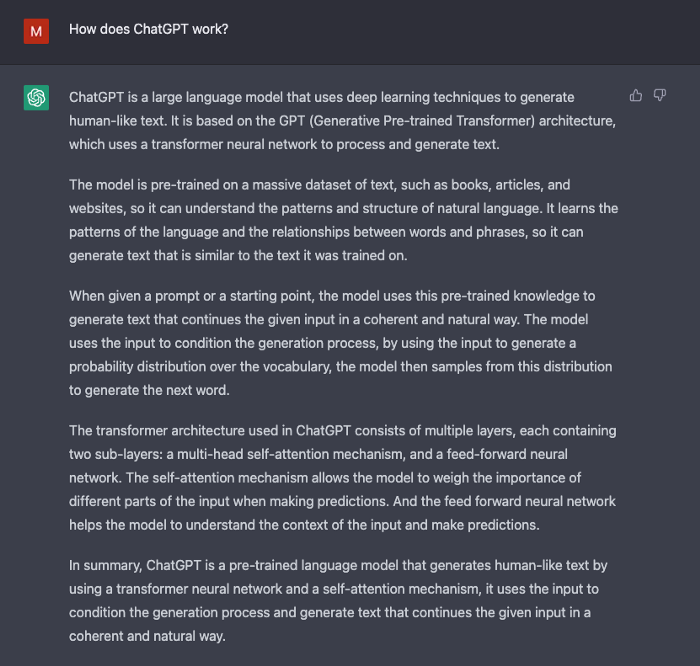
**Helpfulness:** the model’s ability to infer and follow user instructions. Labelers preferred outputs from InstructGPT over GPT-3 85 ± 3% of the time.

**Truthfulness**: the model’s tendency for hallucinations. The PPO model produced outputs that showed minor increases in truthfulness and informativeness when assessed using the [TruthfulQA](https://arxiv.org/abs/2109.07958) dataset.

**Harmlessness**: the model’s ability to avoid inappropriate, derogatory, and denigrating content. Harmlessness was tested using the RealToxicityPrompts dataset. The test was performed under three conditions.

1. Instructed to provide respectful responses: resulted in a significant decrease in toxic responses.
2. Instructed to provide responses, without any setting for respectfulness: no significant change in toxicity.
3. Instructed to provide toxic response: responses were in fact significantly more toxic than the GPT-3 model.

For more information on the methodologies used in creating ChatGPT and InstructGPT, read the original paper published by OpenAI **Training language models to follow instructions with human feedback**, *2022*<https://arxiv.org/pdf/2203.02155.pdf>.



Screenshot from ChatGPT generated by the author.

Happy learning!