✦ Member-only story

How ChatGPT really works, explained for non-technical people

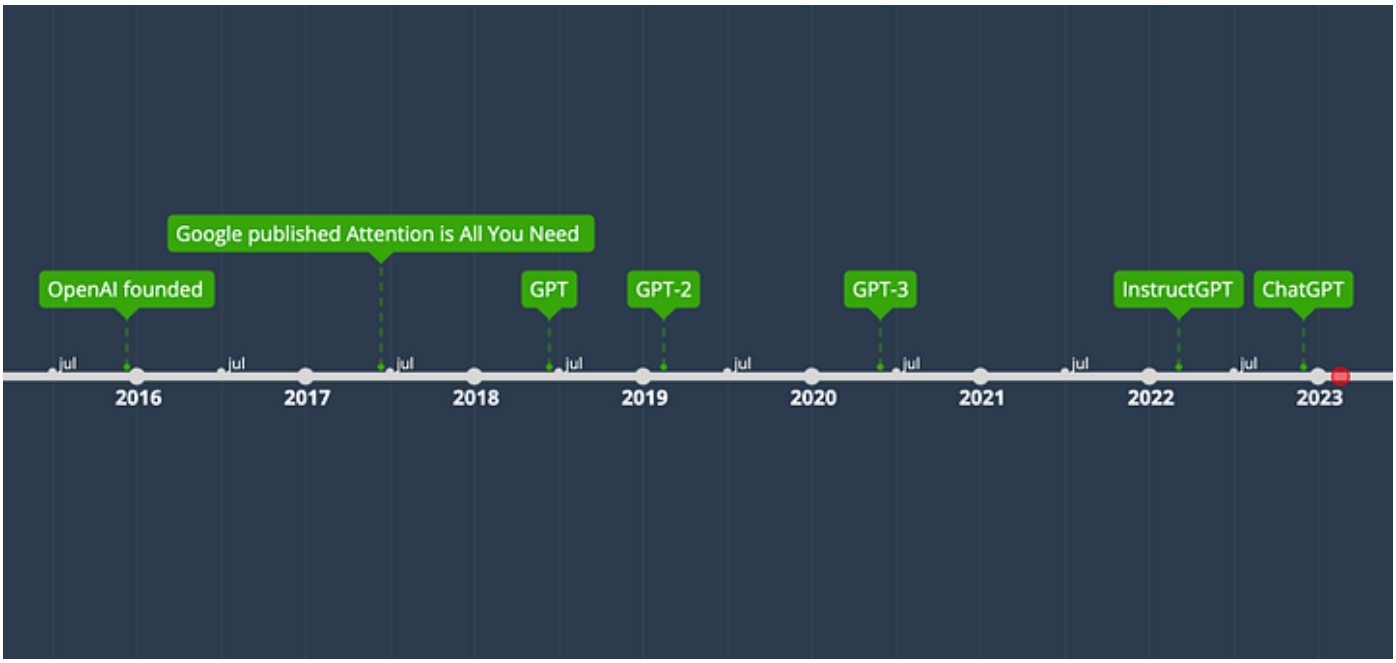
Guodong (Troy) Zhao · Follow
Published in Bootcamp
12 min read · Feb 21

The release of ChatGPT by OpenAI at the end of last year has been phenomenal — even my grandma is asking about it. Its capabilities to generate human-like language have inspired people to experiment with its potential in various products. Its wildly successful launch even put pressure on tech giants like Google to rush to release their own version of ChatGPT.

But let's be honest, for non-technical product managers, designers, and entrepreneurs, the inner workings of ChatGPT may seem like a magical black box. Don't worry! In this blog, I'll try to **explain the technology and the model behind ChatGPT as simply as possible**. By the end of this post, you'll have a good understanding of what ChatGPT can do, and how it performs its magic.

The transformer & GPT timeline

Before we dive deep into the actual mechanism of ChatGPT, let's take a quick look at the timeline of the development of the transformer architecture of language models and the different versions of GPT, so that you can have a better sense of how things evolved into the ChatGPT we have today.



Timeline of Transformers, GPT, and ChatGPT. Image by the author.

- 2015. OpenAI was founded by Sam Altman, Elon Musk, Greg Brockman, Peter Thiel, and others. OpenAI develops many different AI models other than GPT.
- 2017. Google published the paper *Attention is All You Need*, which introduced the transformer architecture [2]. The transformer is a neural network architecture that lays the foundation for many state-of-the-art (SOTA) large language models (LLM) like GPT.
- 2018. GPT is introduced in *Improving Language Understanding by Generative Pre-training* [3]. It's based on a modified transformer architecture and pre-trained on a large corpus.
- 2019. GPT-2 is introduced in *Language Models are Unsupervised Multitask Learners* [4], which can perform a range of tasks without explicit supervision when training.
- 2020. GPT-3 is introduced in *Language Models are Few-Shot Learners* [5], which can perform well with few examples in the prompt without fine-tuning.
- 2022. InstructGPT is introduced in *Training language models to follow instructions with human feedback* [6], which can better follow user instructions by fine-tuning with human feedback.

- 2022. ChatGPT, a sibling of InstructGPT, is introduced in *ChatGPT: Optimizing Language Models for Dialogue*. It can interact with humans in conversations, thanks to the fine-tuning with human examples and reinforcement learning from human feedback (RLHF).

The timeline shows that GPT evolved from the original transformer architecture and gained its ability through many iterations. If you don't understand the terms like transformers, pre-training, fine-tuning, or reinforcement learning from human feedback, no worries! I'll explain them all in the next sections.

Diving deep behind the models

Now that you know ChatGPT is based on transformers and the preceding GPT models, let's take a closer look at the components of those models and how they work. It's okay if you are not familiar with deep learning, neural networks, or AI — I will leave out the equations and explain the concepts using analogies and examples.

In the next sections, I will start with a generalized overview of language models & NLP, moving on to the original transformer architecture, then to how GPT adapted the transformer architecture, and finally how ChatGPT is fine-tuned based on GPT.

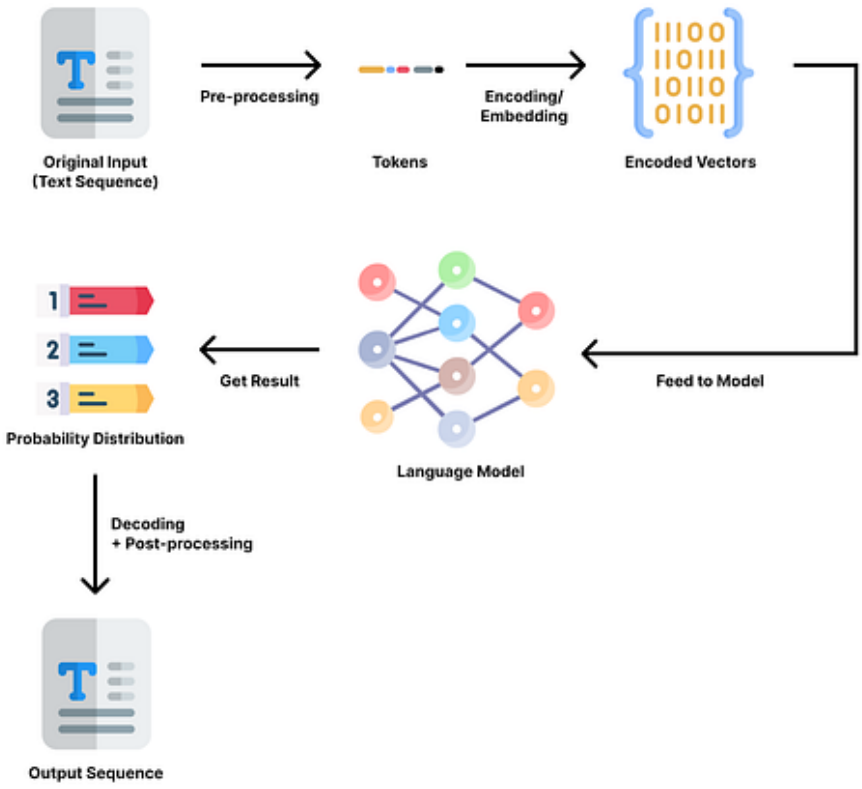
Language models & NLP

There are many types of AI or deep learning models. For natural language processing (NLP) tasks like conversations, speech recognition, translation, and summarization, we will turn to language models to help us.

Language models can learn a library of text (called *corpus*) and **predict words or sequences of words with probabilistic distributions, i.e. how likely a word or sequence can occur**. For example, when you say "Tom likes to eat ...", the probability of the next word being "pizza" would be higher than "table". If it's predicting the next word in the sequence, it's called next-token-prediction; if it's predicting a missing word in the sequence, it's called masked language modeling.

Since it's a probability distribution, there can be many probable words with different probabilities. Although you might think it's ideal to always choose the best candidate with the highest probability, it may lead to repetitive sequences. So in practice,

researchers would add some randomness (**temperature**) when choosing the word from the top candidates.



Typical NLP Process with a Language Model. Image by the author.

In a typical NLP process, the input text will go through the following steps [8]:

- **Preprocessing**: cleaning the text with techniques like sentence segmentation, tokenization (breaking down the text into small pieces called **tokens**), stemming (removing suffixes or prefixes), removing stop words, correcting spelling, etc. For example, “Tom likes to eat pizza.” would be tokenized into [“Tom”, “likes”, “to”, “eat”, “pizza”, “.”] and stemmed into [“Tom”, “like”, “to”, “eat”, “pizza”, “.”].
- **Encoding or embedding**: turn the cleaned text into a vector of numbers, so that the model can process.
- **Feeding to model**: pass the encoded input to the model for processing.
- **Getting result**: get a result of a probability distribution of potential words represented in vectors of numbers from the model.

- **Decoding**: translate the vector back to human-readable words.
- **Post-processing**: refine the output with spell checking, grammar checking, punctuation, capitalization, etc.

AI researchers have come up with many different model architectures. Transformers, a type of neural network, has been state-of-the-art for recent years and lays the foundation for GPT. In the next section, we'll look at transformers' components and mechanisms.

Transformer architecture

The transformer architecture is the foundation for GPT. It is a type of neural network, which is similar to the neurons in our human brain. The transformer can understand contexts in sequential data like text, speech, or music better with mechanisms called **attention** and **self-attention**.

Attention allows the model to **focus on the most relevant parts of the input and output by learning the relevance or similarity between the elements**, which are usually represented by vectors. If it focuses on the same sequence, it's called self-attention [2] [9].

The attention mechanism measures the relevance/similarity between each element. Image by the author.

Let's take the following sentence as an example: "Tom likes to eat apples. He eats them every day." In this sentence, "he" refers to "Tom" and "them" refers to "apples". And the attention mechanism uses a mathematical algorithm to tell the model that those words are related by calculating a similarity score between the word vectors. With this

mechanism, transformers can better “make sense” of the meanings in the text sequences in a more coherent way.

Transformers have the following components [2]:

- **Embedding & Positional Encoding**: turning words into vectors of numbers
- **Encoder**: extract features from the input sequence and analyze the meaning and context of it. It outputs a matrix of hidden states for each input token to be passed to the decoder
- **Decoder**: generate the output sequence based on the output from the encoder and the previous output tokens
- **Linear & Softmax Layer**: turning the vector into a probability distribution of output words

The encoder and decoder are the main components of transformer architecture. **The encoder is responsible for analyzing and “understanding” the input text and the decoder is responsible for generating output.**

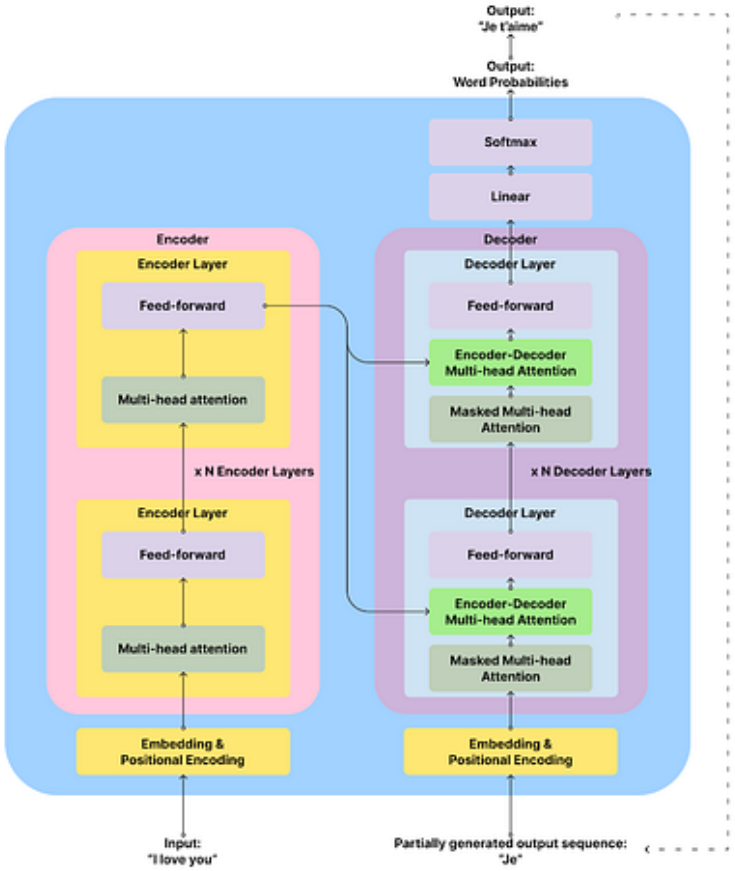


Illustration by Author. Adapted from Attention is All You Need [2]

If you're interested, you can read on for the details of encoders and decoders. If not, feel free to jump to the next section where we'll cover GPT, which is a variant of transformers.

The encoder is a stack of multiple identical layers (6 in the original transformer paper). Each layer has two sub-layers: a multi-head self-attention layer and a feed-forward layer, with some connections, called residual connection and layer normalization [2]. The multi-head self-attention sub-layer applies the attention mechanism to find the connection/similarity between input tokens to understand the input. The feed-forward sub-layer does some processing before passing the result to the next layer to prevent overfitting. **You can think of encoders like reading books — you will pay attention to each new word you read and think about how it's related to the previous words.**

The decoder is similar to the encoder in that it's also a stack of identical layers. But each decoder layer has an additional encoder-decoder attention layer between the self-attention and feed-forward layers, to allow the decoder to attend to the input sequence.

For example, if you're translating "I love you" (input) to "Je t'aime" (output), you will need to know "Je" and "I" are aligned and "love" and "aime" are aligned.

The multi-head attention layers in the decoder are also different. They're masked to not attend to anything to the right of the current token, which has not been generated yet [2]. **You can think of decoders like free-form writing — you write based on what you've written and what you've read, without caring about what you're going to write.**

From transformers to GPT, GPT2, and GPT3

GPT's full name is **Generative Pre-trained Transformer**. From the name, you can see that it's a generative model, good at generating output; it's pre-trained, meaning it has learned from a large corpus of text data; it's a type of transformer.

In fact, GPT uses only the decoder part of the transformer architecture [3]. From the previous transformers section, we learned that decoders are responsible for predicting the next token in the sequence. **GPT repeats this process again and again by using the previously generated results as input to generate longer texts**, which is called **auto-regressive.** For example, if it's translating "I love you" to French, it will first generate "Je", then use the generated "Je" to get "Je t'aime". (See the dashed line in the previous illustration).

In training the first version of GPT, researchers used unsupervised pre-training with the BookCorpus database, consisting of over 7000 unique unpublished books [3]. **Unsupervised learning is like having the AI read those books itself and try to learn the general rules of language and words**. On top of the pre-training, they also used supervised fine-tuning on specific tasks like summarization or question and answering. **Supervised means that they will show the AI examples of requests and correct answers and ask the AI to learn from those examples.**

In GPT-2, researchers expanded the size of the model (1.5B parameters) and the corpus they feed to the model with WebText, which is a collection of millions of web pages, during the unsupervised pre-training [4]. With such a big corpus to learn from, the model proved that it can perform very well on a wide range of language related-tasks even without supervised fine-tuning.

In GPT-3, the researchers took a step further in expanding the model to 175 billion parameters and using a huge corpus comprising hundreds of billions of words from the web, books, and Wikipedia. With such a huge model and a big corpus in pre-training, researchers found that **GPT-3 can learn to perform tasks better with one (one-shot) or a few examples (few-shot) in the prompt without explicit supervised fine-tuning**.

(If you want to learn more about how to prompt the models to produce better results, you can read my other article: How to use ChatGPT in product management)

At this stage, the GPT-3 model is already impressive. But they're more like general-purpose language models. Researchers wanted to explore how it can follow human instructions and have conversations with humans. Therefore, they created InstructGPT and ChatGPT based on the general GPT model. Let's see how they did it in the next section.

Teaching GPT to interact with humans: InstructGPT and ChatGPT

After the iterations from GPT to GPT-3 with growing models and corpus size, researchers realized that bigger models don't mean that they can follow human intent well and may produce harmful outputs. Therefore, they attempted to fine-tune GPT-3 with **supervised learning** and **reinforcement learning from human feedback (RLHF)** [6] [12]**.** With these training steps came the two fine-tuned models — InstructGPT and ChatGPT.

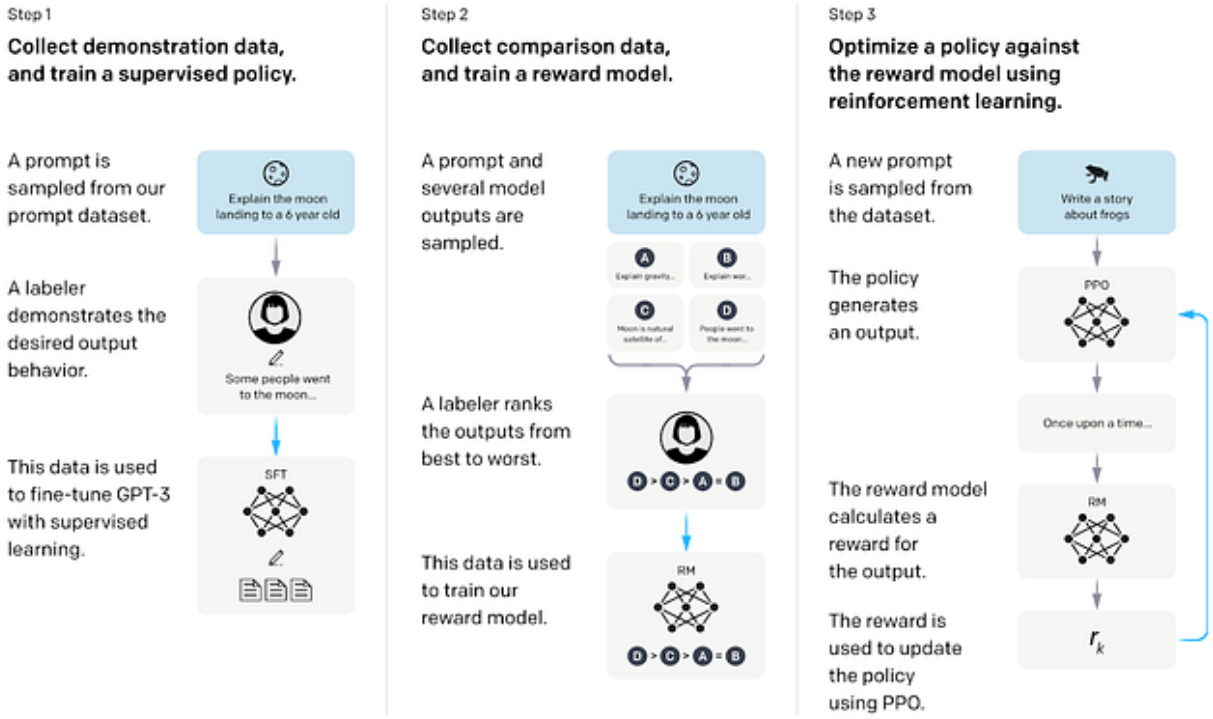


Illustration from Training language models to follow instructions with human feedback [6]

The first step is **supervised learning from human examples**. Researchers first provided the pre-trained GPT with a curated, labeled dataset of prompt and response pairs written by human labelers. This dataset is used to let the model learn the desired behavior from those examples. From this step, they get a supervised fine-tuned (SFT) model.

The second step is **training a reward model (RM) to rate the responses from the generative model**. Researchers used the SFT model to generate multiple responses from each prompt and asked human labelers to rank the responses from best to worse by quality, engagement, informativeness, safety, coherence, and relevance. The prompts, responses, and rankings are fed to a reward model to learn human preferences of the responses through supervised learning. The reward model can predict a scalar reward value based on how well the response matches human preferences.

In the third step, **researchers used the reward model to optimize the SFT model's policy through reinforcement learning**. The SFT model will generate a response from a new prompt; the reward model will assess the response and give it a reward value that approximates human preference; the reward is then used to optimize the generative

model by updating its parameters. For example, if the generative model generates a response that the reward model thinks humans may like, it will get a positive reward to continue generating similar responses in the future; and vice versa.

Through this process with supervised learning and reinforcement learning from human feedback, the InstructGPT model (with only 1.3B parameters) is able to perform better in tasks that follow human instructions than the much bigger GPT-3 model (with 175 B parameters).

(Note: However, this does not mean that InstructGPT is better than GPT-3 in all aspects or domains. For example, GPT-3 may still have an advantage in generating longer or more creative texts, such as stories or articles.)

ChatGPT is a sibling model to InstructGPT. The training process is similar for ChatGPT and InstructGPT, including the same methods of supervised learning and RLHF we covered earlier. The main difference is that ChatGPT is trained with examples from conversational tasks, like question answering, chit-chat, trivia, etc [7]. Through this training, ChatGPT can have natural conversations with humans in dialogues. In conversations, ChatGPT can answer follow-up questions and admit mistakes, making it more engaging to interact with.

(You can read this article to learn different ways to further tailor foundation language models like GPT for your business)

Takeaways

With the previous explanations, I hope you have a clearer understanding of how the model behind ChatGPT works and how it evolved into what it is today.

As a quick recap, here are the most important takeaways and the limitations of GPT models:

- ChatGPT is based on a decoder-only auto-regressive transformer model to **take in a sequence of text and output a probability distribution of tokens in the sequence**, generating one token at a time iteratively.
- Since it doesn't have the ability to search for references in real-time, it's making probabilistic predictions in the generation process based on the corpus it has been

trained on, which can lead to false claims about facts.

- It's **pre-trained on a huge corpus of web and book data and fine-tuned with human conversation examples** through supervised learning and reinforcement learning from human feedback (RLHF).
- Its capability is mainly based on its model size and the quality and size of the corpus and examples it learned from. With some additional supervised learning or RLHF, it can perform better in specific contexts or tasks.
- Since the corpus comes from web content and books, **they might have biases that the model can learn from**, especially for social, cultural, political, or gender-based biases, resulting in biased responses to some requests.

Stay curious and open to new technologies like ChatGPT. I believe an open mindset and curiosity can help us non-technical product managers, designers, and entrepreneurs navigate this new wave of technological revolution.

(Update: GPT-4 is announced in March, 2023. Here is what you need to know about GPT-4 and its business implications)

(Knowing how the GPT works can be a big step in productizing it, but there might be way more problems during the implementation of a GPT/LLM-backed app, read Cookbook for solving common problems in building GPT/LLM apps if you want to know how to solve them.)

References

[1] "Introducing OpenAI." *OpenAI*, 12 Dec. 2015, https://openai.com/blog/introducing-openai/.

[2] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

[3] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

[4] Radford, Alec, et al. "Language models are unsupervised multitask learners." *OpenAI blog* 1.8 (2019): 9.

[5] Brown, Tom, et al. “Language models are few-shot learners.” *Advances in neural information processing systems* 33 (2020): 1877–1901.

[6] Ouyang, Long, et al. “Training language models to follow instructions with human feedback.” *arXiv preprint arXiv:2203.02155* (2022).

[7] “ChatGPT: Optimizing Language Models for Dialogue.” *OpenAI*, 30 Nov. 2022, https://openai.com/blog/chatgpt/.

[8] Murali, Aishwarya. “A Guide to Perform 5 Important Steps of NLP Using Python.” *Analytics Vidhya*, 17 Aug. 2021, https://www.analyticsvidhya.com/blog/2021/08/a-guide-to-perform-5-important-steps-of-nlp-using-python/.

[9] Cristina, Stefania. “The Transformer Attention Mechanism.” *Machine Learning Mastery*, 15 Sept. 2022, https://machinelearningmastery.com/the-transformer-attention-mechanism/.

[10] Doshi, Ketan. “Transformers Explained Visually (Part 1): Overview of Functionality.” *Medium*, 3 June 2021, https://towardsdatascience.com/transformers-explained-visually-part-1-overview-of-functionality-95a6dd460452.

[11] Kosar, Vaclav. *Feed-Forward, Self-Attention & Key-Value*. 2 Jan. 2021, https://vaclavkosar.com/ml/Feed-Forward-Self-Attendion-Key-Value-Memory.

[12] “Aligning Language Models to Follow Instructions.” *OpenAI*, 27 Jan. 2022, https://openai.com/blog/instruction-following/.

AI | Product Management | ChatGPT | Machine Learning | Tech