

Chapter 2. Introduction to Large Language Models for Text Generation

A NOTE FOR EARLY RELEASE READERS

With Early Release ebooks, you get books in their earliest form—the authors' raw and unedited content as they write—so you can take advantage of these technologies long before the official release of these titles.

This will be the 2nd chapter of the final book. Please note that the GitHub repo will be made active later on.

If you have comments about how we might improve the content and/or examples in this book, or if you notice missing material within this chapter, please reach out to the author at ccollins@oreilly.com.

In artificial intelligence, a recent focus has been the evolution of large language models (LLMs). Unlike their less-flexible predecessors, LLMs are capable of handling and learning from a much larger volume of data, resulting in the emergent capability of producing text that closely resembles human language output. These models have generalized across diverse applications, from writing content, to automating software development and enabling real-time interactive chatbot experiences.

What Are Text Generation Models?

Text generation models utilize advanced algorithms to understand the meaning in text and produce outputs that are often indistinguishable from human work. If you've ever interacted with [ChatGPT](#) or marveled at its ability to craft coherent and contextually relevant sentences, you've witnessed the power of an LLM in action.

In natural language processing (NLP) and LLMs, the fundamental linguistic unit is a *token*. [Tokens](#) can represent sentences, words, or even sub-words such as a set of characters. A useful way to understand the size of text data is by looking at the number of tokens it comprises; for instance, a text of 100 tokens roughly equates to about 75 words. This comparison can be essential for managing the processing limits of LLMs as different models may have varying token capacities.

Tokenization, the process of breaking down text into tokens, is a crucial step in preparing data for NLP tasks. Several methods can be used for tokenization, including [Byte-Pair Encoding](#) (BPE), WordPiece, and SentencePiece. Each of these methods has its unique advantages and is suited to particular use cases. BPE is commonly used due to its efficiency in handling a wide range of vocabulary while keeping the number of tokens manageable.

BPE begins by viewing a text as a series of individual characters. Over time, it combines characters that frequently appear together into single units, or tokens. To understand this better, consider the word *apple*. Initially, BPE might see it as *a*, *p*, *p*, *l*, and *e*. But after noticing that *p* often comes after *a* and before *l* in the dataset, it might combine them and treat *apl* as a single token in future instances.

This approach helps LLMs recognize and generate words or phrases, even if they weren't common in the training data, making the models more adaptable and versatile.

Understanding the workings of LLMs requires a grasp of the underlying mathematical principles that power these systems. Although the computations can be complex, we can simplify the core elements to provide an intuitive understanding of how these models operate. Particularly within a business context, the accuracy and reliability of LLMs are paramount.

A significant part of achieving this reliability lies in the pre-training and fine-tuning phases of LLM development. Initially, models are trained on vast datasets during the pre-training phase, acquiring a broad understanding of language. Subsequently, in the fine-tuning phase, models are adapted for specific tasks, honing their capabilities to provide accurate and reliable outputs for specialized applications.

Vector Representations: The Numerical Essence of Language

In the realm of NLP words aren't just alphabetic symbols. They can be tokenized and then represented in a numerical form, known as *vectors*. These vectors are multi-dimensional arrays of numbers that capture the semantic and syntactic relations:

$$w \rightarrow \mathbf{v} = [v_1, v_2, \dots, v_n]$$

Creating word vectors, also known as *word embeddings* relies on intricate patterns within language. During an intensive training phase, models are designed to identify and learn these patterns, ensuring that words with similar meanings are mapped close to one another in a high-dimensional space ([Figure 2-1](#)).

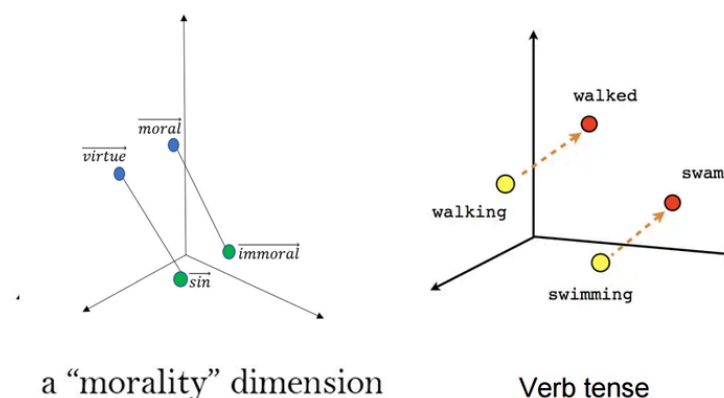


Figure 2-1. Semantic proximity of word vectors within a word embedding space

The beauty of this approach is its ability to capture nuanced relationships between words and calculate their distance. When we examine word embeddings, it becomes evident that words with similar or related meanings

like *virtue* and *moral* or *walked* and *walking* are situated near each other. This spatial closeness in the embedding space becomes a powerful tool in various NLP tasks, enabling models to understand context, semantics, and the intricate web of relationships that form language.

Transformer Architecture: Orchestrating Contextual Relationships

Before we delve deep into the mechanics of transformer architectures, let's build a foundational understanding. In simple terms, when we have a sentence, say, *The cat sat on the mat*, each word in this sentence gets converted into its numerical vector representation. So, *cat* might become a series of numbers, as does *sat*, *on*, and *mat*.

As you'll explore in detail later in this chapter, the transformer architecture takes these word vectors and understands their relationships – both in structure (syntax) and meaning (semantics). There are many types of transformers, [Figure 2-2](#) showcases both BERT and GPT's architecture. Additionally, transformers don't just see words in isolation, it looks at *cat* and knows it's related to *sat* and *mat* in a specific way in this sentence.

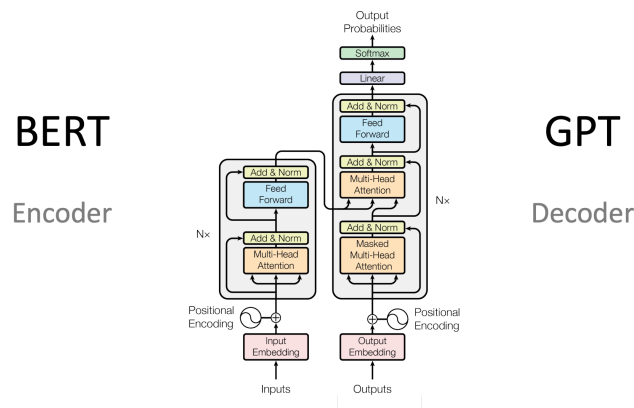


Figure 2-2. BERT uses an encoder for input data, while GPT has a decoder for output

When the transformer processes these vectors, it uses mathematical operations to understand the relationships between the words, thereby producing new vectors with rich, contextual information:

$$\mathbf{v}'_i = \text{Transformer}(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m)$$

One of the remarkable features of transformers is their ability to comprehend the nuanced contextual meanings of words. The [self-attention](#) mechanism in transformers lets each word in a sentence look at all other words to understand its context better. Think of it like each word casting votes on the importance of other words for its meaning. By considering the entire sentence, transformers can more accurately determine the role and meaning of each word, making their *interpretations more contextually rich*.

Probabilistic Text Generation: The Decision Mechanism

After the transformer understands the context of the given text, it moves on to generating new text, guided by the concept of likelihood or probability. In mathematical terms, the model calculates how likely each possi-

ble next word is to follow the current sequence of words, and picks the one that is most likely:

$$w_{next} = \text{argmax } P(w|w_1, w_2, \dots, w_m)$$

By repeating this process, as shown in [Figure 2-3](#), the model generates a coherent and contextually relevant string of text as its output.

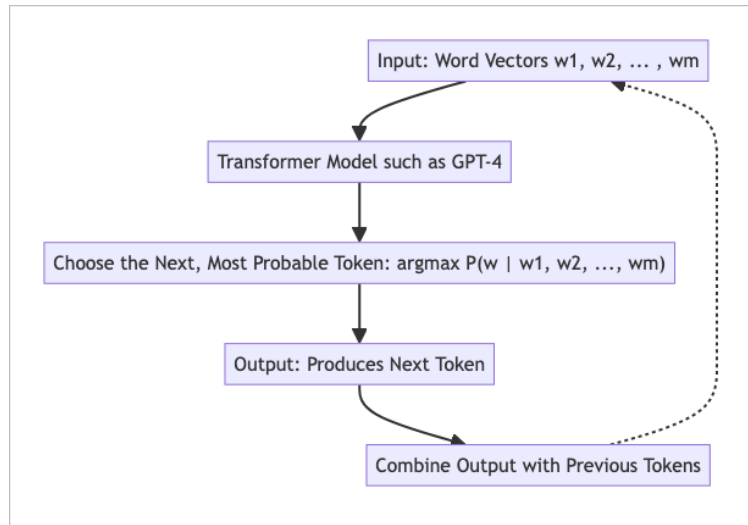


Figure 2-3. How text is generated using transformer models such as GPT-4

The mechanisms driving LLMs are rooted in vector mathematics, linear transformations, and probabilistic models. While the under-the-hood operations are computationally intensive, the core concepts are built on these mathematical principles, offering a foundational understanding that bridges the gap between technical complexity and business applicability.

Historical Underpinnings: The Rise of Transformer Architectures

Language models like ChatGPT, the *GPT* in which stands for *generative pre-trained transformer*, didn't magically emerge. They're the culmination of years of progress in the field of NLP, with particular acceleration since the late 2010s. At the heart of this advancement is the introduction of transformer architectures, which were detailed in the groundbreaking paper [Attention Is All You Need](#) by the Google Brain team.

The real breakthrough of transformer architectures was the concept of *attention*. Traditional models processed text sequentially, which limited their understanding of language structure especially over long distances of text. Attention transformed this by allowing models to directly relate distant words to one another irrespective of their positions in the text. This was a groundbreaking proposition. It meant that words and their context didn't have to move through the entire model to affect each other. This not only significantly improved the models' text comprehension but also made them much more efficient.

This attention mechanism played a vital role in expanding the models' capacity to detect long-range dependencies in text. This was crucial for generating outputs that were not just contextually accurate and fluent, but also coherent over longer stretches.

According to AI pioneer and educator [Andrew Ng](#), much of the early NLP research, including the fundamental work on transformers, received significant funding from U.S. military intelligence agencies. Their keen interest in tools like machine translation and speech recognition, primarily for intelligence purposes, inadvertently paved the way for developments that transcended just translation.

Training LLMs requires extensive computational resources. These models are fed with vast amounts of data, ranging from terabytes to petabytes, including internet content, academic papers, books, and more niche datasets tailored for specific purposes. It's important to note, however, that the data used to train LLMs can carry *inherent biases from their sources*. Thus, users should exercise caution and ideally employ human oversight when leveraging these models, ensuring responsible and ethical AI applications.

OpenAI's GPT-4, for example, boasts an estimated [1.7 trillion parameters](#). *Parameters* in the context of neural networks are the weights and biases adjusted throughout the training process, allowing the model to represent and generate complex patterns based on the data it's trained on.

To meet these requirements, major technological companies such as Microsoft, Meta, Google have invested heavily making LLM development a high-stakes endeavor.

The rise of LLMs has provided an increased demand for the hardware industry, particularly companies specializing in graphics processing units (GPUs). NVIDIA, for instance has become almost synonymous with high-performance GPUs that are essential for training LLMs.

The demand for powerful, efficient GPUs has skyrocketed as companies strive to build ever-larger and more complex models. It's not just the raw computational power that's sought after. GPUs also need to be fine-tuned for tasks endemic to machine learning, like tensor operations. *Tensors*, in a machine learning context, are multi-dimensional arrays of data, and operations on them are foundational to neural network computations. This emphasis on specialized capabilities has given rise to tailored hardware such as NVIDIA's H100 Tensor Core GPUs, explicitly crafted to expedite machine learning workloads.

Furthermore, the overwhelming demand often outstrips the supply of these top-tier GPUs, sending prices on an upward trajectory. This supply-demand interplay has transformed the GPU market into a fiercely competitive and profitable arena. Here, an eclectic clientele, ranging from tech behemoths to academic researchers, scramble to procure the most advanced hardware.

This surge in demand has sparked a wave of innovation beyond just GPUs. Companies are now focusing on creating dedicated AI hardware, such as Google's Tensor Processing Units (TPUs), to cater to the growing computational needs of AI models.

This evolving landscape underscores not just the symbiotic ties between software and hardware in the AI sphere but also spotlights the ripple effect of the LLM *gold rush*. It's steering innovations and funneling investments into various sectors, especially those offering the fundamental components for crafting these models.

OpenAI's Generative Pre-trained Transformers

Founded with a mission to ensure that artificial general intelligence benefits all of humanity, [OpenAI](#) has recently been at the forefront of the AI revolution. One of their most groundbreaking contributions has been the GPT series of models, which have substantially redefined the boundaries of what LLMs can achieve.

The original GPT model by OpenAI was more than a mere research output; it was a compelling demonstration of the potential of transformer-based architectures. This model showcased the initial steps towards making machines understand and generate human-like language, laying the foundation for future advancements.

The unveiling of GPT-2 was met with both anticipation and caution. Recognizing the model's powerful capabilities, OpenAI initially hesitated in releasing it due to concerns about its potential misuse. Such was the might of GPT-2 that ethical concerns took center stage, which might look quaint compared to the power of today's models. However, when OpenAI decided to [open-source](#) the project, it didn't just mean making the code public. It allowed businesses and researchers to use these pretrained models as building blocks, incorporating AI into their applications without starting from scratch. This move democratized access to high-level natural language processing capabilities, spurring innovation across various domains.

After GPT-2, OpenAI decided to focus on releasing paid, closed-source models. GPT-3's arrival marked a monumental stride in the progression of LLMs. It garnered significant media attention, not just for its technical prowess but also for the societal implications of its capabilities. This model could produce text so convincing that it often became indistinguishable from human-written content. From crafting intricate pieces of literature to churning out operational code snippets, GPT-3 exemplified the seemingly boundless potential of AI.

GPT-3.5-turbo and ChatGPT

Bolstered by Microsoft's significant investment in their company, OpenAI introduced GPT-3.5-turbo, an optimized version of its already exceptional predecessor. Following a [\\$1 billion injection](#) from Microsoft in 2019, which later increased to a hefty \$13 billion for a 49% stake in OpenAI's for-profit arm, OpenAI used these resources to develop GPT-3.5-turbo that offered improved efficiency and affordability, effectively making LLMs more accessible for a broader range of use cases.

OpenAI wanted to gather more world feedback for fine-tuning, and so [ChatGPT](#) was born. Unlike its general-purpose siblings, [ChatGPT](#) was [fine-tuned](#) to excel in conversational contexts, enabling a dialogue between humans and machines that felt natural and meaningful.

[Figure 2-4](#) shows the training process for ChatGPT, which involves three main steps:

- *Collection of demonstration data:* In this step, human labelers provide examples of the desired model behavior on a distribution of prompts. The labelers are trained on the project and follow specific instructions to annotate the prompts accurately.
- *Training a supervised policy:* The demonstration data collected in the previous step is used to fine-tune a pre-trained GPT-3 model using supervised learning. In supervised learning, models are trained on a labeled dataset where the correct answers are provided. This step helps the model to learn to follow the given instructions and produce outputs that align with the desired behavior.
- *Collection of comparison data and reinforcement learning:* In this step, a dataset of model outputs is collected, and human labelers rank the outputs based on their preference. A reward model is then trained to predict which outputs the labelers would prefer. Finally, reinforcement learning techniques, specifically the Proximal Policy Optimization (PPO) algorithm, are used to optimize the supervised policy to maximize the reward from the reward model.

This training process allows the ChatGPT model to align its behavior with human intent. The use of reinforcement learning with human feedback helped create a model that is more helpful, honest, and safe compared to the pretrained GPT-3 model.

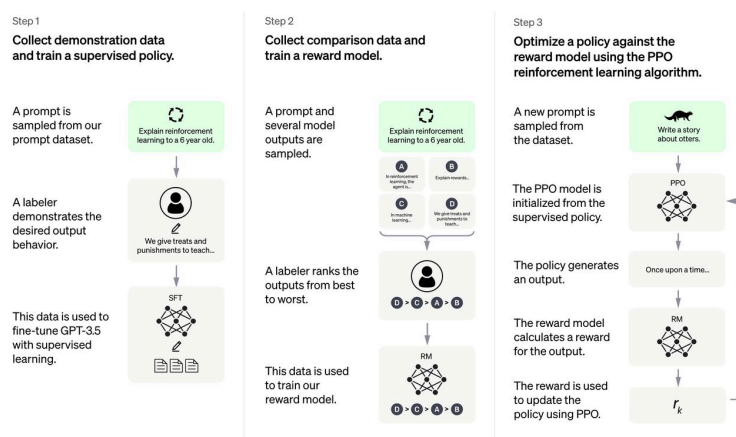


Figure 2-4. The fine-tuning process for ChatGPT

According to a [UBS study](#), by January 2023 ChatGPT set a new benchmark, amassing 100 million active users and becoming the fastest-growing consumer application in internet history. ChatGPT is now a go-to for customer service, virtual assistance, and numerous other applications that require the finesse of human-like conversation.

GPT-4

In 2024, OpenAI released GPT-4 which excels in understanding complex queries and generating contextually relevant and coherent text. For example, GPT-4 scored in the 90th percentile of the bar exam with a score of 298 out of 400. Currently, GPT-3.5-turbo is free to use in ChatGPT, but GPT-4 requires a [monthly payment](#).

GPT-4 uses a [mixture of experts](#) approach; it goes beyond relying on a single model's inference to produce even more accurate and insightful results.

Google's Gemini

After Google lost search market share due to ChatGPT usage, it initially released Bard on March 21st, 2023. Bard was a bit [rough around the edges](#), and definitely didn't initially have the same high quality LLM responses that ChatGPT offered ([Figure 2-5](#)).

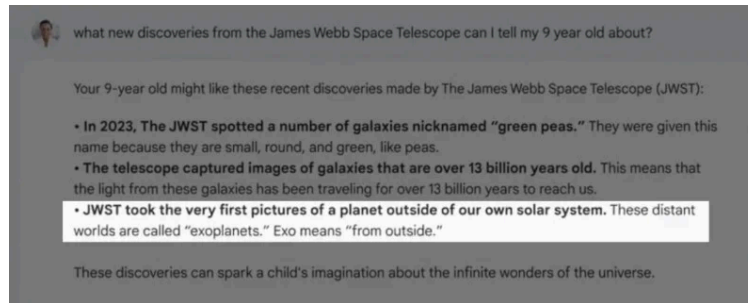


Figure 2-5. Bard hallucinating results about the James Webb Space Telescope

Google has kept adding extra features over time including code generation, visual AI, real-time search, and voice into Bard, bringing it closer to ChatGPT in terms of quality.

On March 14th, 2023 Google released [PaLM API](#), allowing developers to access it on Google Cloud Platform. In April 2023 AWS released similar services such as [Amazon Bedrock](#) and [Amazon's Titan FMs](#). Google [re-branded Bard to Gemini](#) for their v1.5 release in February 2024, and started to get results similar to GPT-4.

Also, Google released two smaller [open source models](#) based on the same architecture as Gemini. OpenAI is finally no longer the only obvious option for software engineers to integrate state-of-the-art large language models into their applications.

Meta's LLaMA and Open Source

Meta's approach to language models differs significantly from other competitors in the industry. By sequentially releasing open-source models [LLaMa](#) and [LLaMa2](#), Meta aims to foster a more inclusive and collaborative AI development ecosystem.

The open-source nature of LLaMa and [LLaMa2](#) has significant implications for the broader tech industry, especially for large enterprises. The transparency and collaborative ethos encourage rapid innovation, as problems and vulnerabilities can be quickly identified and addressed by the global developer community. As these models become more robust and secure, large corporations can adopt them with increased confidence.

Meta's open-source strategy not only democratizes access to state-of-the-art AI technologies but also has the potential to make a meaningful impact across the industry. By setting the stage for a collaborative, transparent, and decentralized development process, LLaMa and LLaMa2 are pioneering models that could very well define the future of generative AI. The models are available in 7 billion, 13 billion, and 70 billion parameter versions on Amazon Web Services (AWS), Hugging Face, and other platforms.

The open-source nature of these models presents a double-edged sword. On one hand, it levels the playing field. This means that even smaller developers have the opportunity to contribute to innovation, improve and apply open source models to practical business applications. This kind of decentralized innovation could lead to breakthroughs that might not occur within the walled gardens of a single organization, enhancing the model's capabilities and applications.

However, the same openness that makes this possible also poses potential risks, as it could allow malicious actors to exploit this technology for detrimental ends. This indeed is a concern that organizations like OpenAI share, suggesting that some degree of control and restriction can actually serve to mitigate the dangerous applications of these powerful tools.

Leveraging Quantization and LoRA

One of the game-changing aspects of these open-source models is the potential for [quantization](#) and the use of [LoRA](#) (low-rank approximations). These techniques allow developers to fit the models into smaller hardware footprints. Quantization helps to reduce the numerical precision of the model's parameters, thereby shrinking the overall size of the model without a significant loss in performance. Meanwhile, LoRA assists in optimizing the network's architecture, making it more efficient to run on consumer-grade hardware.

Such optimizations make fine-tuning these LLMs increasingly feasible on consumer hardware. This is a critical development because it allows for greater experimentation and adaptability. No longer confined to high-powered data centers, individual developers, small businesses, and startups can now work on these models in more resource-constrained environments.

Mistral

Mistral 7B, a brainchild of French startup [Mistral AI](#), emerges as a powerhouse in the generative AI domain, with its 7.3 billion parameters making a significant impact. This model is not just about size; it's about efficiency and capability, promising a bright future for open-source large language models and their applicability across a myriad of use cases. The key to its efficiency is the implementation of sliding window attention, a technique released under a permissive Apache open-source license, many AI engineers have fine-tuned on top of this model as a base, including the impressive [Zephyr 7b beta](#) model. There is also [Mixtral 8x7b](#), a mixture of experts model (similar to the architecture of GPT-4), which achieves results similar to GPT-3.5-turbo.

For a more detailed and up-to-date comparison of open-source models and their performance metrics, visit the Chatbot [Arena Leaderboard](#) hosted by Hugging Face.

Anthropic: Claude

Released on July 11, 2023, [Claude 2.0](#) is setting itself apart from other prominent LLMs such as ChatGPT and LLaMA, with its pioneering [Constitutional AI](#) approach to AI safety and alignment – training the model using a list of rules or values. A notable enhancement in Claude 2.0 was its expanded context window of 100k tokens, and the ability to upload files. In the realm of generative AI, a *context window* refers to the amount of text or data the model can actively consider or keep in mind when generating a response. With a larger context window, the model can understand and generate based on a broader context.

This advancement garnered significant enthusiasm from AI engineers, as it opened up avenues for new and more intricate use cases. For instance, Claude 2.0's augmented ability to process more information at once makes it adept at summarizing extensive documents or sustaining in-depth conversations. The advantage was short-lived, as OpenAI released their 128k version of GPT-4 only [six months later](#), however the fierce competition between rivals is pushing the field forward.

GPT-4V(ision)

In a significant leap forward on September 23, 2023, OpenAI expanded the capabilities of GPT-4 with the introduction of Vision, enabling users to instruct GPT-4 to analyze images alongside text. This innovation was also reflected in the update to ChatGPT's interface, which now supports the inclusion of both images and text as user inputs. This development signifies a major trend towards *multi-modal models*, which can seamlessly process and understand multiple types of data, such as images and text, within a single context.

Model Comparison

The market for LLMs is dominated by OpenAI at the time of writing, with its state-of-the-art GPT-4 model widely considered to have a significant lead. The closest competitor is Anthropic, and there is widespread excitement at the potential of smaller open-source models such as LLaMA and Mistral, particularly with respects to fine-tuning. Although commentators expect OpenAI to continue to deliver world-beating models in the future, as open-source models get *good enough* at more tasks, AI workloads may shift towards local fine-tuned models. With advances in model performance and quantization (methods for trading off accuracy versus size and compute cost) it may be possible to one day run LLMs on your mobile phone or other devices. For now, the best way to get a sense for what the models are good at is to run the same prompt across multiple models and compare the responses. One thing that regularly stands out in our work is that GPT-4 is much better at following instructions, as is demonstrated in the following example where it was the only model to respond in the right format, with names that matched the examples (starting with the letter i), as desired.

Input:

```
Brainstorm a list of product names for a shoe that fits any foot size, in the style of Steve Jobs.

Return the results as a comma separated list, in this format:
Product description: A shoe that fits any foot size
Product names: [list of 3 product names]

## Examples
Product description: A refrigerator that dispenses beer
Product names: iBarFridge, iFridgeBeer, iDrinkBeerFridge

Product description: A watch that can tell accurate time in space
Product names: iNaut, iSpace, iTime

Product description: A home milkshake maker
Product names: iShake, iSmoothie, iShake Mini
```

Output (GPT-4):

```
Product description: A shoe that fits any foot size
Product names: iFit, iShoe, iFlexible
```

Output (Claude 2):

```
Product description: A shoe that fits any foot size

Product names: iFit, OneSizeFitsAll, AdaptiShoe
```

Output (LLaMA 2 13b):

```
Sure! Here are three product name ideas for a shoe that fits any foot size, in the style of Steve Jobs.

Product description: A shoe that fits any foot size, with innovative technology and sleek design.

Product names:

1. iFit: A shoe that adapts to your foot, no matter the size.
2. iMorph: A shoe that shapes to your foot for a perfect fit.
3. iFlex: A shoe that adjusts to your foot's unique contours for maximum comfort.

Each of these names incorporates elements of innovation, technology, and flexibility, which are
```

Summary

The journey of LLMs from their inception to their current advanced states is a tale of relentless innovation, collaboration, and a bit of competition. As these models continue to evolve, they are likely to become even more integral parts of our daily lives, changing the way we interact with technology and even with each other.

By understanding the historical context and capabilities of these models, you can better appreciate the tools at our disposal for various applications, from prompt engineering to the development of intelligent virtual agents.

As we move forward, the landscape of LLMs promises not just technological advancement, but an ever-expanding range of possibilities for human-

machine interaction. It's important to note, however, that while these models offer expansive possibilities, data privacy remains a crucial concern. If these models use your data for retraining or fine-tuning, exercise caution and refrain from inputting sensitive information.

In the next chapter, you will delve into the fascinating world of basic prompt engineering techniques for working with text LLMs. You'll learn the essential skills needed to get the most out of powerful language models like GPT-4. Exciting insights and practical methods await you as you unlock the true potential of generative AI.