

✦ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



# Introduction to Large Language Models and the Transformer Architecture



Pradeep Menon · [Follow](#)

7 min read · Mar 8, 2023



230



ChatGPT is making waves worldwide, attracting over 1 million users in record time. As a CTO for startups, I discuss this revolutionary technology daily due to the persistent buzz and hype surrounding it. The applications of GPT are limitless, but only some take the time to understand how these models work. This blog post aims to demystify OpenAI's GPT (Generative Pre-trained Transformer) language model.

GPT (Generative Pre-trained Transformer) is a type of language model that has gained significant attention in recent years due to its ability to perform various natural languages processing tasks, such as text generation, summarization, and question-answering.

This blog post will explore the fundamental concept of LLMs (Large Language Models) and the transformer architecture, which is the building block of all language models with transformers, including GPT. By the end of this post, you will have a basic understanding of the building blocks of a Large Language Model, such as GPT.

Let's start by understanding what Large Language Models (LLMs) are.

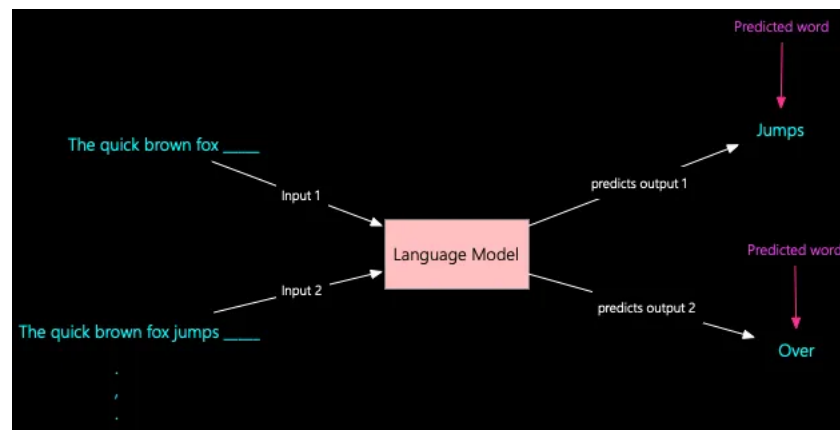
## Large Language Models (LLM)

Large Language Models (LLMs) are trained on massive amounts of text data. As a result, they can generate coherent and fluent text. LLMs perform well on various natural languages processing tasks, such as language translation, text summarization, and conversational agents. LLMs perform so well because they are pre-trained on a large corpus of text data and can be fine-

tuned for specific tasks. GPT is an example of a Large Language Model. These models are called “large” because they have billions of parameters that shape their responses. For instance, GPT-3, the largest version of GPT, has 175 billion parameters and was trained on a massive corpus of text data.

The basic premise of a language model is its ability to predict the next word or sub-word (called tokens) based on the text it has observed so far. To better understand this, let’s look at an example.

Top highlight

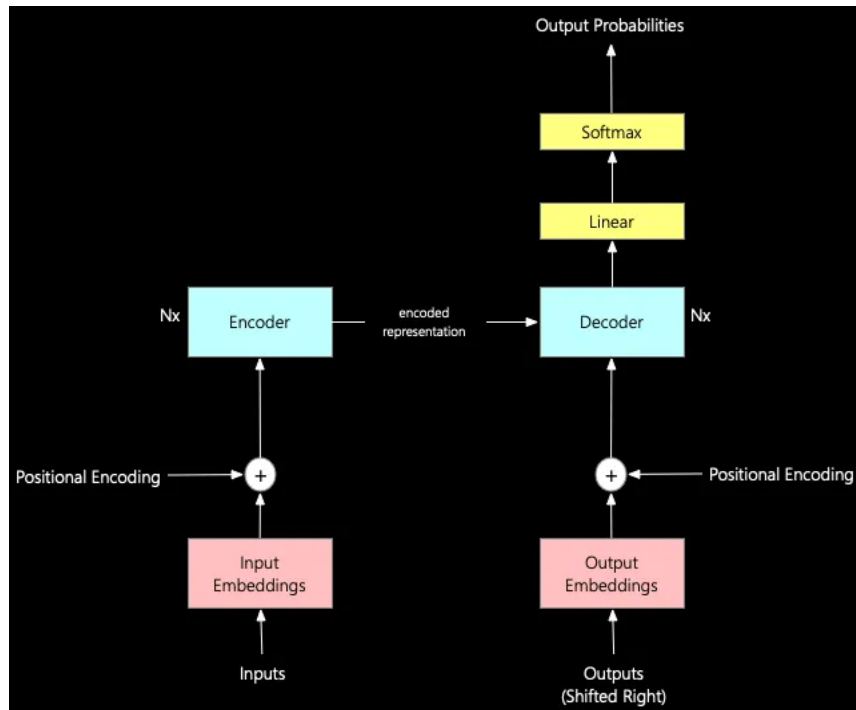


The above example shows that the language model predicts one token at a time by assigning probabilities to tokens based on its training. Typically, the token with the highest probability is used as the next part of the input. This process is repeated continuously until a special <stop> token is selected.

The deep learning architecture that has made this process more human-like is the Transformer architecture. So let us now briefly understand the Transformer architecture.

## The Transformer Architecture: The Building Block

The transformer architecture is the fundamental building block of all Language Models with Transformers (LLMs). The transformer architecture was introduced in the paper “Attention is all you need,” published in December 2017. The simplified version of the Transformer Architecture looks like this:



There are seven important components in transformer architecture. Let's go through each of these components and understand what they do in a simplified manner:

1. **Inputs and Input Embeddings:** The tokens entered by the user are considered inputs for the machine learning models. However, models only understand numbers, not text, so these inputs need to be converted into a numerical format called "input embeddings." Input embeddings represent words as numbers, which machine learning models can then process. These embeddings are like a dictionary that helps the model understand the meaning of words by placing them in a mathematical space where similar words are located near each other. During training, the model learns how to create these embeddings so that similar vectors represent words with similar meanings.
2. **Positional Encoding:** In natural language processing, the order of words in a sentence is crucial for determining the sentence's meaning. However, traditional machine learning models, such as neural networks, do not inherently understand the order of inputs. To address this challenge, positional encoding can be used to encode the position of each word in the input sequence as a set of numbers. These numbers can be fed into the Transformer model, along with the input embeddings. By incorporating positional encoding into the Transformer architecture, GPT can more effectively understand the order of words in a sentence and generate grammatically correct and semantically meaningful output.

3. **Encoder:** The encoder is part of the neural network that processes the input text and generates a series of hidden states that capture the meaning and context of the text. The encoder in GPT first tokenizes the input text into a sequence of tokens, such as individual words or sub-words. It then applies a series of self-attention layers; think of it as voodoo magic to generate a series of hidden states that represent the input text at different levels of abstraction. Multiple layers of the encoder are used in the transformer.
4. **Outputs (shifted right):** During training, the decoder learns how to guess the next word by looking at the words before it. To do this, we move the output sequence over one spot to the right. That way, the decoder can only use the previous words. With GPT, we train it on a ton of text data, which helps it make sense when it writes. The biggest version, GPT-3, has 175 billion parameters and was trained on a massive amount of text data. Some text corpora we used to train GPT include the Common Crawl web corpus, the BooksCorpus dataset, and the English Wikipedia. These corpora have billions of words and sentences, so GPT has a lot of language data to learn from.
5. **Output Embeddings:** Models can only understand numbers, not text, like input embeddings. So the output must be changed to a numerical format, known as “output embeddings.” Output embeddings are similar to input embeddings and go through positional encoding, which helps the model understand the order of words in a sentence. A loss function is used in machine learning, which measures the difference between a model’s predictions and the actual target values. The loss function is particularly important for complex models like GPT language models. The loss function adjusts some parts of the model to improve accuracy by reducing the difference between predictions and targets. The adjustment ultimately improves the model’s overall performance, which is great! Output embeddings are used during both training and inference in GPT. During training, they compute the loss function and update the model parameters. During inference, they generate the output text by mapping the model’s predicted probabilities of each token to the corresponding token in the vocabulary.
6. **Decoder:** The positionally encoded input representation and the positionally encoded output embeddings go through the decoder. The decoder is part of the model that generates the output sequence based on the encoded input sequence. During training, the decoder learns how to guess the next word by looking at the words before it. The decoder in GPT generates natural language text based on the input sequence and the context learned by the encoder. Like an encoder, multiple layers of

decoders are used in the transformer.

7. **Linear Layer and Softmax:** After the decoder produces the output embeddings, the linear layer maps them to a higher-dimensional space. This step is necessary to transform the output embeddings into the original input space. Then, we use the softmax function to generate a probability distribution for each output token in the vocabulary, enabling us to generate output tokens with probabilities.

## The Concept of Attention Mechanism

Attention is all you need.

The transformer architecture beats out other ones like Recurrent Neural networks (RNNs) or Long short-term memory (LSTMs) for natural language processing. The reason for the superior performance is mainly because of the “attention mechanism” concept that the transformer uses. The attention mechanism lets the model focus on different parts of the input sequence when making each output token.

- The RNNs don't bother with an attention mechanism. Instead, they just plow through the input one word at a time. On the other hand, Transformers can handle the whole input simultaneously. Handling the entire input sequence, all at once, means Transformers do the job faster and can handle more complicated connections between words in the input sequence.
- LSTMs use a hidden state to remember what happened in the past. Still, they can struggle to learn when there are too many layers (a.k.a. the vanishing gradient problem). Meanwhile, Transformers perform better because they can look at all the input and output words simultaneously and figure out how they're related (thanks to their fancy attention mechanism). Thanks to the attention mechanism, they're really good at understanding long-term connections between words.

Let's summarize:

- It lets the model selectively focus on different parts of the input sequence instead of treating everything the same way.
- It can capture relationships between inputs far away from each other in the sequence, which is helpful for natural language tasks.
- It needs fewer parameters to model long-term dependencies since it only

has to pay attention to the inputs that matter.

- It's really good at handling inputs of different lengths since it can adjust its attention based on the sequence length.

## Conclusion

This blog post provides a primer to Large Language Models (LLMs) and the Transformer Architecture that powers LLMs like GPT. Large Language Models (LLMs) have revolutionized natural language processing by providing models that generate coherent and fluent text. They are pre-trained on massive amounts of text data and can be fine-tuned for specific tasks. The Transformer architecture is the fundamental building block of all LLMs. It has made it possible for models like GPT to generate more accurate and contextually relevant output. With the ability to perform various natural language processing tasks, such as text generation, summarization, and question-answering, LLMs like GPT are opening up new possibilities for communication and human-machine interaction. In short, LLMs have greatly improved natural language processing. They have the potential to enhance human-machine interaction in exciting ways.

## References

1. [Attention is all you need](#)
2. [Introducing ChatGPT \(openai.com\)](#)
3. [Transformers, Explained: Understand the Model Behind GPT-3, BERT, and T5 \(daleonai.com\)](#)
4. [What are Transformer Neural Networks? — YouTube](#)
5. [Stanford Webinar — GPT-3 & Beyond](#)

[Gpt 3](#)[NLP](#)[OpenAI](#)[Machine Learning](#)



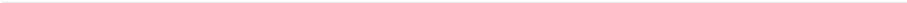
Written by Pradeep Menon

4K Followers

Follow



Creating impact through Technology | #CTO at #Microsoft| Data & AI Strategy | Cloud Computing | Design Thinking | Blogger | Public Speaker | Published Author



More from Pradeep Menon

Pradeep Menon in Towards Data Science

### Data Science Simplified Part 7: Log-Log Regression Models

In the last few blog posts of this series, we discussed simple linear regression model. We

7 min read · Aug 14, 2017

130

2

Pradeep Menon in Towards Data Science

### Data Science Simplified Part 5: Multivariate Regression Models

In the last article of this series, we discussed the story of Fernando. A data scientist who


8 min read · Aug 5, 2017

570

17

Pradeep Menon

### A Deep-Dive into Fine-Tuning of Large Language Models

 New blog on Fine-Tuning LLMs like GPT-4 & BERT! Dive deep into tailoring AI. Insights,

6 min read · Aug 13, 2023

57

1

Pradeep Menon

### Mastering Generative AI Interactions: A Guide to In-Context

OpenAI's GPT (Generative Pre-trained Transformer) models have revolutionized the

12 min read · Apr 18, 2023

18


See all from Pradeep Menon

Recommended from Medium

https://rpradeepmenon.medium.com/introduction-to-large-language-models-and-the-transformer-architecture-534408ed7e61

Page 8 of 10




 Amanatullah

## Transformer Architecture explained

Transformers are a new development in machine learning that have been making a lot

10 min read · Sep 1, 2023

 510

 5





 Mark Riedl

## A Very Gentle Introduction to Large Language Models without the

1. Introduction

38 min read · Apr 13, 2023

 7.1K

 118






### Lists




**Predictive Modeling w/ Python**  
20 stories · 956 saves




**Practical Guides to Machine Learning**  
10 stories · 1128 saves



**Natural Language Processing**  
1244 stories · 725 saves



**The New Chatbots: ChatGPT, Bard, and Beyond**  
12 stories · 320 saves

 Andreas Stöffelbauer in Data Science at Microsoft

How Large Language Models Work

From zero to ChatGPT

25 min read · Oct 24, 2023

 719

 10





 Shaw Talebi in Towards Data Science


How to Build an LLM from Scratch

Data Curation, Transformers, Training at Scale, and Model Evaluation

★ · 16 min read · Sep 21, 2023

 1.5K

 10





 Mastering LLM (Large Language Model)

LLM Training: A Simple 3-Step Guide You Won't Find Anywhere


Discover How Language Models are Trained in 3 Easy Steps


6 min read · Oct 1, 2023

 248

 3





 Simohamed Amara

what the difference between gpt vs LLM ?

GPT (Generative Pre-trained Transformer) and LLM (Language Model) are terms that are

★ · 4 min read · Dec 25, 2023

 86







See more recommendations