GPT-2

What is a Language Model?

In The Illustrated Word2Vec, we've looked at what a language model is – basically a machine learning model that is able to look at part of a sentence and predict the next word. The most famous language models are smartphone keyboards that sugest the next word based on what you've currently typed.

In this sense, we can say that the GPT-2 is basically the next word prediction feature of a keyboard app, but one that is much larger and more sophisticated than what your phone has. The GPT-2 was trained on a massive 40GB dataset called Web-Text that the OpenAI researchers crowld from the internet as part of the researchers effort. To compare in terms of storage size, the keyboard app I use, Swiftkey, takes up 78MB of space. The smallest variant of the trained GPT2, takes up 500MBs of storage of store all its parameter. The largest GPT-2 variant is 13 times the size so it could take up more than 6.5 GBs of storage space.

GPT-2 Small
↓
117M parameter

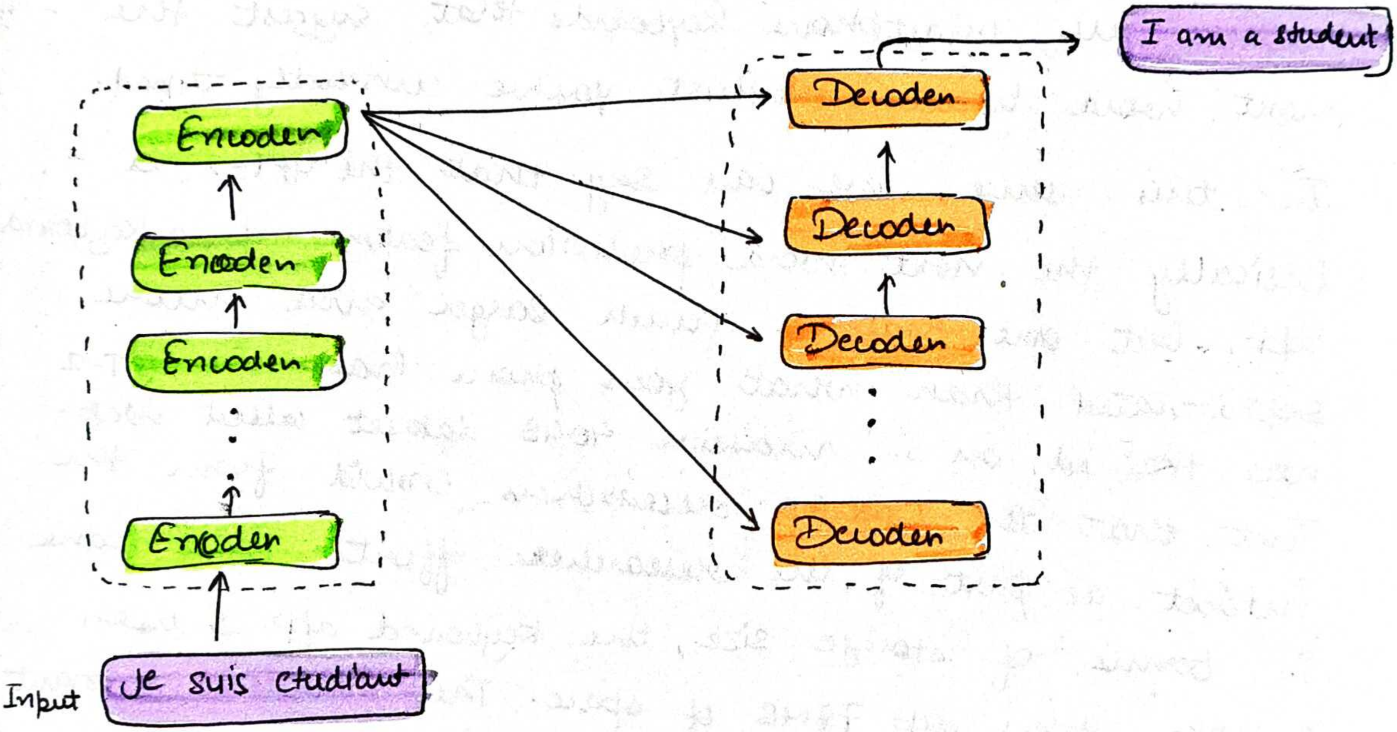
GPT-2 Medium
↓
345M parameter

GPT-2 Large
↓
762M Parameter

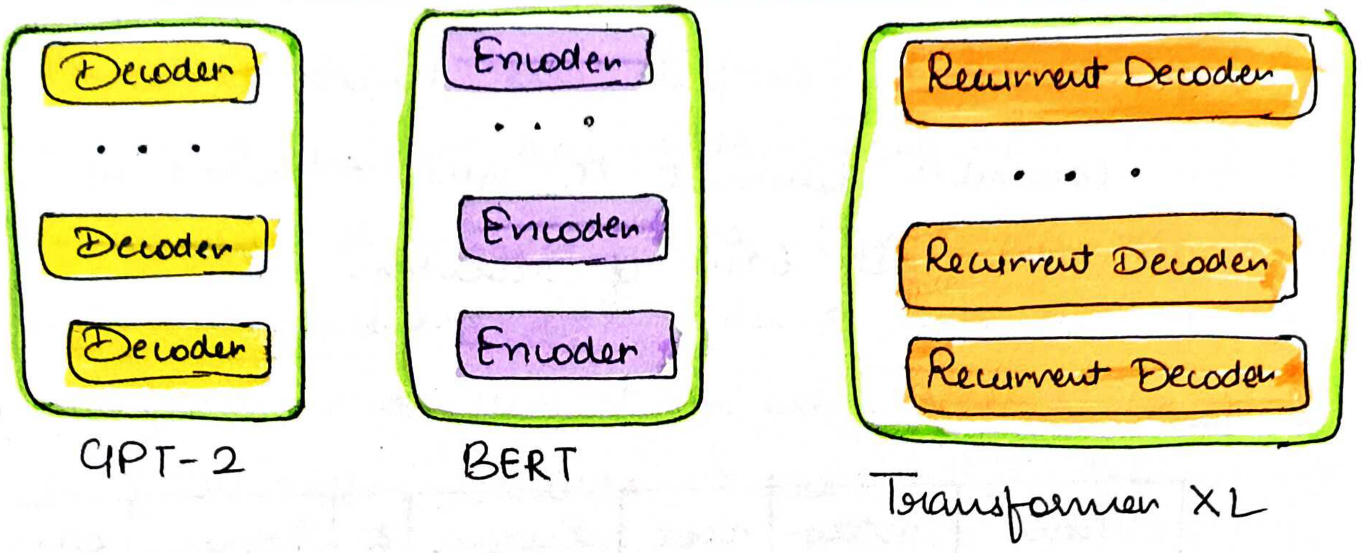
GPT-2 Extra Large
↓
1,542M parameter

Transformer for Language Modeling

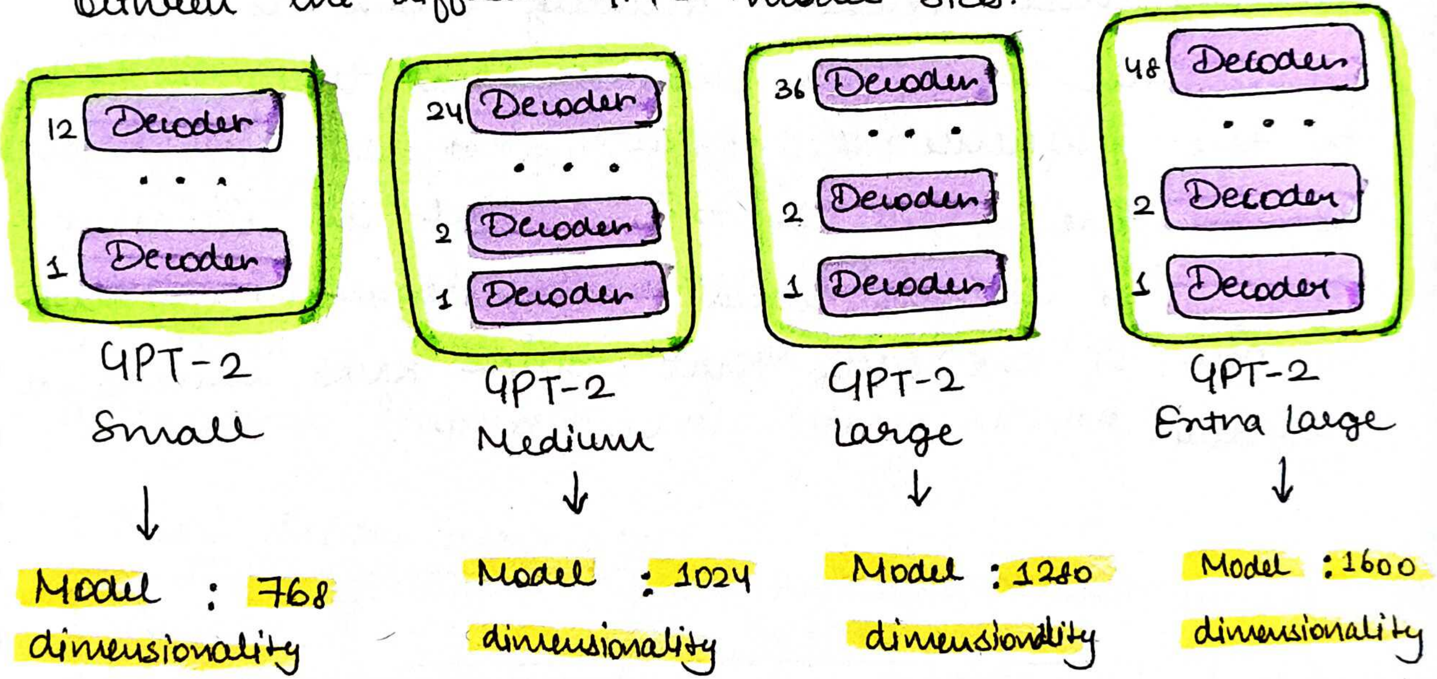
The original transformer model is made up of an encoder and decoder each is a stack of what we can call transformer blocks. That architecture was appropriate because the model tackled machine translation - a problem where encoder - decoder architectures have been successful in the past.



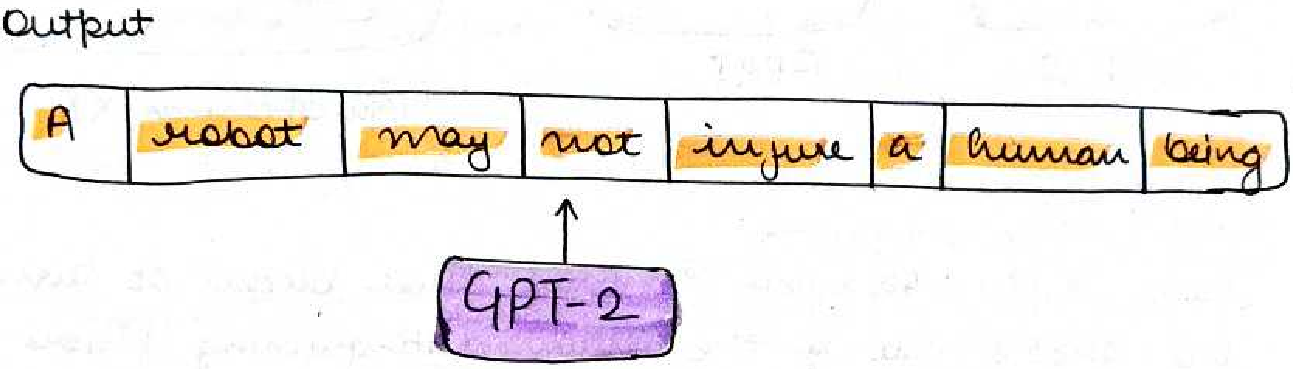
A lot of the subsequent research work saw the architecture shed either the encoder or decoder, and use just one stack of transformer blocks - stacking them up as higer as practically possible, feeding them massive amounts of training text, and throwing vast amounts of compute at them (hundreds of dollars to train some of these language models, likely millions in the case of AlphaStar).



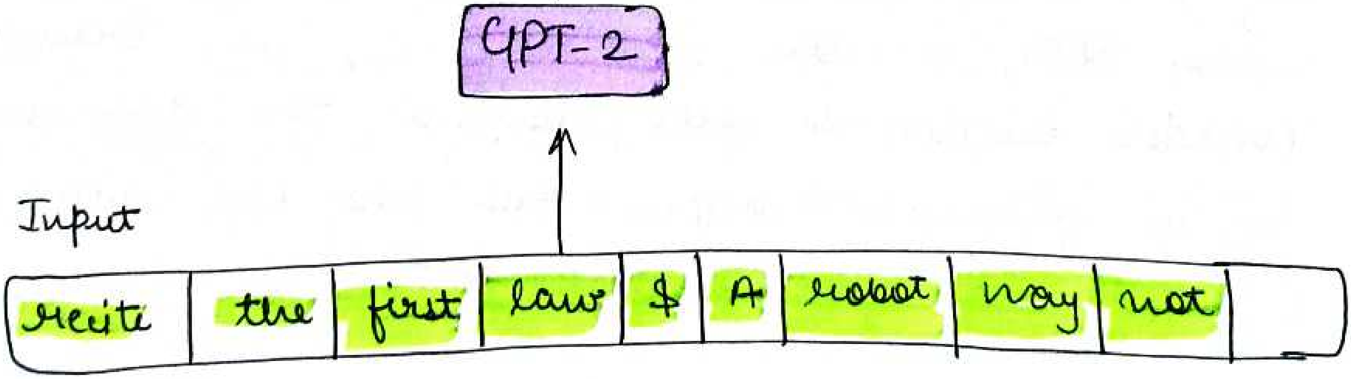
How higher can we stack up these blocks? It turns out that's one of the main distinguishing factors between the different GPT2 model sizes:



The GPT-2 is built using transformer decoder blocks. BERT, on the other hand, uses transformer encoder blocks. We will examine the difference in a following section. But one key difference between the two is that GPT2, like traditional language models, output one token at a time. Let's for example prompt a well-trained GPT-2 to recite the first law of robotics:



The way these models actually work is that after each token is product, that token is added to the sequence of inputs. And the new sequence becomes the Input to the model in its next step. This is an called "auto-regression". This is one of the ideas that made RNNs unreasonably effective.



The GPT2, and some later models like Transform XL and XLNet are auto-regression in nature. BERT is not. That is a trade off. In losing auto-regression, BERT gained the ability to incorporate the content on both sides of a word to gain better results. XLNet brings back auto-regression while finding an alternative way to incorporate the content on both sides.

Architecture of GPT-2 is same as Architecture of GPT

Difference between GPT and GPT-2

The difference betn GPT and GPT-2 lies primarily in their scale, training data, capabilities and architecture improvements. Here's detailed comparison:

1. Model Scale and Size:

Feature	GPT	GPT-2
Parameters	~117M	Range from 117M to 1.58B (small, medium, large, XL)
Model sizes	Single model	Multiple sizes released (124M, 355M, 774M, 1.5B)

- GPT-2 is significantly larger than GPT, with its largest variant containing 1.5 billion parameters, enabling it to capture more complex patterns and dependencies.

2. **Training Data:**

- **GPT:** Trained on BooksCorpus (a dataset of 11,000 books)

- **GPT-2:** Trained on a much larger and diverse datasets called WebText, which contains approximately 8 million high-quality web pages (filtered for language quality).

The larger and more diverse training data gives GPT-2 better generalization and performance across various tasks.

3. **Performance:**

- **GPT:**

  → Performs well but is limited in generalization and coherence for complex tasks.

  → Requires more fine-tuning to achieve good results on specific applications.

- **GPT-2:**

  → Significantly better at generating coherent, contextually relevant and longer text.

→ Can perform many NLP tasks zero-shot (without task-specific fine-tuning), making it more versatile.

4. **Zero-shot and Few-shot Capabilities:**

- **GPT:** Limited zero-shot capabilities; it often requires fine-tuning for each task.

- **GPT-2:** Remarkable zero-shot and few-shot learning abilities, allowing it to perform well even without task-specific fine-tuning.

5. **Architecture**

- Both GPT and GPT-2 use a decoder-only transformer architecture with unidirectional attention, meaning the model predicts the next token based on past tokens.

- **Improvements in GPT-2:**
  → Increased model depth (number of layers).
  → Enhanced scalability with larger hidden states and attention heads.

6. **Tokenization**

Both use byte pair encoding (BPE) for tokenization, but GPT-2 processes a more extensive vocabulary (around 50,000 tokens) to handle diverse and complex language better.

7. Release Policy:

- GPT: Released entirely to the public, including the model, training data, and code.

- GPT-2: Initially, only smaller versions were released due to concerns about misuse (e.g., generating spam or fake news). The full 1.5B parameter version was later released after furture evaluation.

8. Applications:

- GPT: Early exploration in text generation, summarization, translation, question answering, and more, with stronger performance is general purpose NLP tasks.

- GPT-2: Widely used in creative text generation, summarization, translation, question answering, and more, with stronger performance in general-purpose NLP task.