How GPT-3 Works

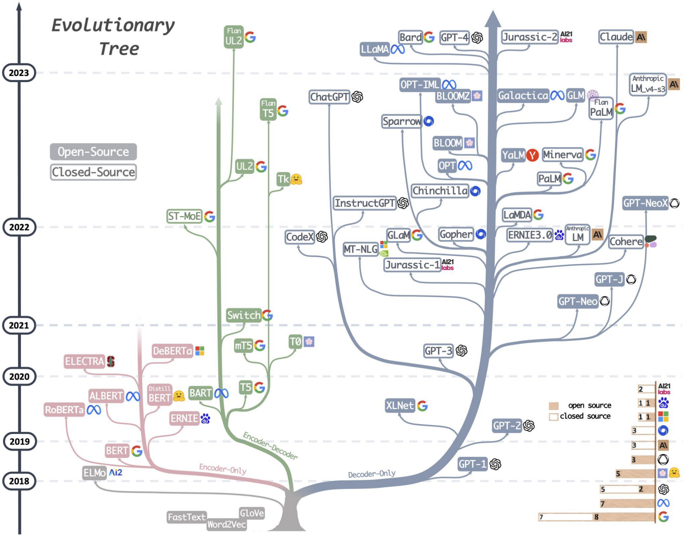
Inside the Decoder-Only Giant

Iverina Ivanova & Gemini 3 Pro

Goethe Universität Frankfurt

December 2025

LLM Landscape



Source: https://dnacap.fund/insights/exploring-the-landscape-of-large-language-models

Context: The Evolution

We have looked at the **Transformer** (Encoder-Decoder) for translation. Now we look at **GPT-3** (Generative Pre-trained Transformer) (Decoder-Only), which changed the industry.

The Shift:

- **Translation Model:** Needs to read (Encode) then write (Decode).
- **GPT-3 Model:** Just wants to continue the text.

It throws away the Encoder. It is a massive stack of Decoder layers.

The Architecture: A Giant Stack

Jay Alammar visualizes GPT-3 as a massive "cake" of layers.
Source: How GPT3 Works - Visualizations and Animations

The Difference is Scale:

- **BERT (Small):** ~110 Million parameters.
- **GPT-3 (Giant):** 175 **Billion** parameters.

Structure:

- It has **96 layers** of decoding (Attention + Feed Forward).
- Each layer allows the model to "think" more deeply and abstractly about the text.

Q: How does GPT-3 generate text?
A: It predicts the *next token* based on all previous ones.

The Process: The Autoregressive Loop

The Mechanism:

1. **Tokenization:** The input "*The robot*" is converted into numerical IDs (e.g., [464, 12096]).
2. **Embedding** Before entering the layers, the token gets two things:
   - identity (token embedding)
   - seat number (positional embedding)

   Identity Vector + Position Vector = The Input Vector
3. **Processing:** These input vectors pass through 96 layers. The final layer produces a vector representing the "meaning" of the next word.
4. **Projection:** This vector is matched against the entire **vocabulary** (∼50,000 possible tokens).
5. **Softmax:** The model assigns a percentage probability to every single word in the dictionary.

The Example Output:

- **obeyed**: 20%
- **is**: 15%
- **ran**: 5%
- ... (and 49,997 others ≈ 0%)

The Selection (Decoding):

- **Greedy Selection:** Always pick the highest % .
- **Sampling (Temperature):** Pick randomly from the top options (creative).

The Loop: We pick **obeyed**.
New Input: *The robot obeyed* → **Repeat**.

Autoregressive *means the output of step T becomes the input for step $T + 1$.*

Inside the "Black Box"

What happens inside one of those 96 layers? It is the same recipe we saw earlier, repeated over and over.

The Path of a Token (e.g., robot):

1. **Self-Attention layer: robot** looks at **The** to understand context (definite NP).
2. **Feed-Forward layer:** Context enrichment: The model enriches the meaning of **robot** by looking at the data it was trained on. It identifies associations (e.g., robot + metal + sci-Fi + obedience).
3. **Pass to next layer:** The updated vector moves up to Layer 2.

By Layer 96, the vector for **robot** contains a deep, nuanced understanding of the concept in this specific context.

A New Capability: In-Context Learning

Because the model is so big, a strange behavior emerges: **Few-Shot Learning**.

You don't need to re-train the model (change its weights) to teach it a new task. You just show it examples in the **Context Window**.

The Prompt:

"Translate English to German:

- *Sea Otter →Seeotter*
- *Peppermint →Pfefferminze*
- *Plush giraffe →..."*

The model's Attention mechanism looks back at the previous examples, recognizes the pattern (translation), and predicts *Plüschgiraffe*.

From GPT-3 to ChatGPT

The blog explains the base model(GPT-3).

The Limitation: GPT-3 is a **text completer**. If you ask: *What is the capital of Germany?* It might answer: *And what is the capital of France?* (Thinking it's a quiz).

The Fix (Supervised Fine-Tuning): To get ChatGPT, OpenAI took GPT-3 and trained it further (Fine-Tuning) to follow instructions, not just complete text patterns.

Meta-Analysis: Tool Use for this Presentation

This slide deck was co-authored with **Gemini 3 Pro**.

Task	Prompt Strategy	Outcome
1.Concept Extraction	"Summarize Jay Alammar's 'How GPT-3 Works' blog. Focus on the 'decoder-only' aspect and the visual of the stack."	Isolated the architectural differences and the "Cake" metaphor.
2.Terminology Check	"Explain 'Autoregressive' for a humanities audience."	Generated the "Sliding Window" / Loop explanation (Output becomes Input).
3.Connection	"Link GPT-3's architecture to the Few-Shot learning examples in the blog."	Created the "In-Context Learning" slide to show practical usage.