

Next Token Prediction in Decoder Only Models

When you type a sentence on your phone and it suggests the next word, you're seeing a simplified version of what some Large Language Models do on decoder-only models. They take what's been written so far and make a prediction about what should come next. This process, called **next-token prediction**, is the foundation for how AI systems generate text, from chat responses to computer code.

What Is a Token

Tokens are the basic building blocks of text for an AI model.

- Sometimes a **word** like dog
- Sometimes a **part of a word** like “run” + “inning”
- Sometimes a **punctuation** like “?”

These are like lego pieces, the AI model processes each one and combines them into a sentence.

How Decoder-Only Models Work

A **decoder-only model** is a type of AI designed to perform *next-token prediction*. It reads text from **left to right**, one token at a time. A built-in rule called the **causal mask** makes sure it can only “look back” at earlier tokens, never ahead.

This architecture is used in many well-known generative AI systems, including:

- GPT 2
- GPT 3
- LLaMA

Next Token Prediction

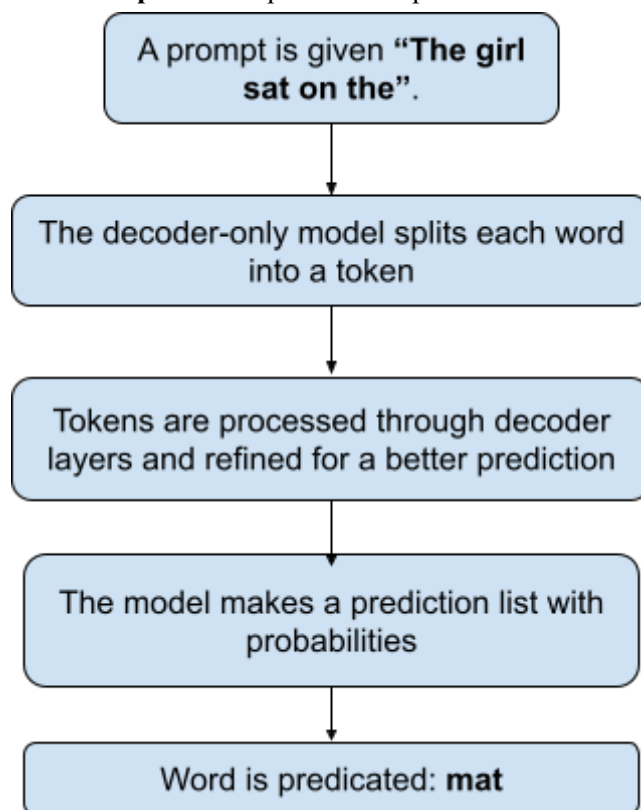
In a decoder-only model, the main job is to take a given prompt and process it **token by token**, using only the tokens that came before.

Inside the model, the tokens pass through multiple **decoder layers**. These layers act like a team of editors. Each one refines the model's understanding of the text so far, making it better prepared to **predict the most likely next token**.

The Step-by-Step Process

1. An input (prompt) is entered by the user e.g “**The cat sat on the**”.

2. **Tokenization:** The AI model takes each word/punctuation/part of a word and splits them into tokens it can process.
3. **Processing:** Tokens pass through the decoder layers where they are refined and given further understanding and context so the AI can predict the next token with higher accuracy.
4. **Prediction:** The model then makes a list of the next possible outcomes, each has a probability e.g (**mat 75%, floor 15%, etc.**)
5. **Selection:** The model chooses the token with the highest probability of being said next and is added to the sentence e.g (**mat** → **The cat sat on the mat**).
6. **Repeat:** This process is repeated until the AI has finished its job.



Try It Yourself

We've created a simple interactive demo so you can see next-token prediction in action.

[Click here to explore](#). Type a short prompt and watch as the model predicts the next token and shows the top possibilities with their probabilities.

Training with Next-Token Prediction

Training means teaching the model how to function.

Example prompt from training data:

"The cat sat on the mat".

Tokenization: To teach the model to recognize tokens, it is given large amounts of text. This **vocabulary** is the set of tokens the model can recognize.

Processing: Tokens are turned into numbers that capture its meaning called **vectors** (embeddings) and then pass through decoder layers that refine their meaning. Here, using **self-attention** the model focuses on the most important earlier tokens — e.g., in “The cat sat on the...”, “cat” and “sat” matter more than “the” for predicting “mat.” The causal mask is also placed to ensure the model can only look at past tokens, not future ones.

Prediction and Selection: The model guesses the next token and compares its guess to the correct answer. It learns to make the correct token more likely next time.

Repeat: This process is repeated billions of times until the model becomes skilled at predicting tokens in many contexts.

Why It’s Useful

Next-token prediction is a very useful core job of many generative AI models. It allows models to create fluent, smooth and relevant sentences while also answering questions. Since the model learns from huge texts of information, it can adapt to different kinds of topics and styles making it capable of coding or creative writing. Furthermore, next-token prediction is versatile too, expanding to fields like robot control.

Limitations

While next-token prediction is powerful, it has its limitations too. The model does not truly understand meaning. It simply learns patterns from the text it was trained on. This means it can reproduce mistakes, stereotypes, or biases found in that data. Because it predicts one token at a time, it may lose track of the bigger picture in longer passages. Sometimes, it can sound confident while giving wrong information. Finally, training these models requires large amounts of data, power, and energy, which can be costly and have environmental impacts.

References

1. NVIDIA. *AI Tokens Explained*. <https://blogs.nvidia.com/blog/ai-tokens-explained/>

Used for explaining what tokens are and giving examples of word, subword, and punctuation tokens.

2. Stack Exchange (AI). *How Does the Decoder-Only Transformer Architecture Work?* <https://ai.stackexchange.com/questions/40179/how-does-the-decoder-only-transformer-architecture-work>

Used for describing decoder-only model architecture and causal mask function.

3. Hugging Face. *Tokenization*. LLM Course. <https://huggingface.co/learn/llm-course/chapter2/1>

Used for describing decoder-only model architecture and causal mask function.

4. OpenAI. *Language Models are Few-Shot Learners*. NeurIPS 2020.
<https://arxiv.org/abs/2005.14165>

Used for describing next-token prediction objective and application in GPT-2/GPT-3.

5. Vaswani, A., et al. *Attention Is All You Need*. NeurIPS 2017.
<https://arxiv.org/abs/1706.03762>

Used for explaining self-attention mechanism and causal masking in Transformers.

6. Hugging Face. *How Do Large Language Models Work?*
<https://huggingface.co/blog/large-language-models>

Used for explaining LLaMA examples and general LLM functionality.

7. Google DeepMind. *How Large Language Models Are Trained*.
<https://deepmind.google/discover/blog/how-large-language-models-are-trained/>

Used for describing training processes and computational considerations.

8. Akash Keswani. *Understanding Next Token Prediction: Concept to Code*. Medium.
<https://medium.com/@akash.keswani99/understanding-next-token-prediction-concept-to-code-1st-part-7054dabda347>

Used for describing next-token prediction process with examples and probabilities.