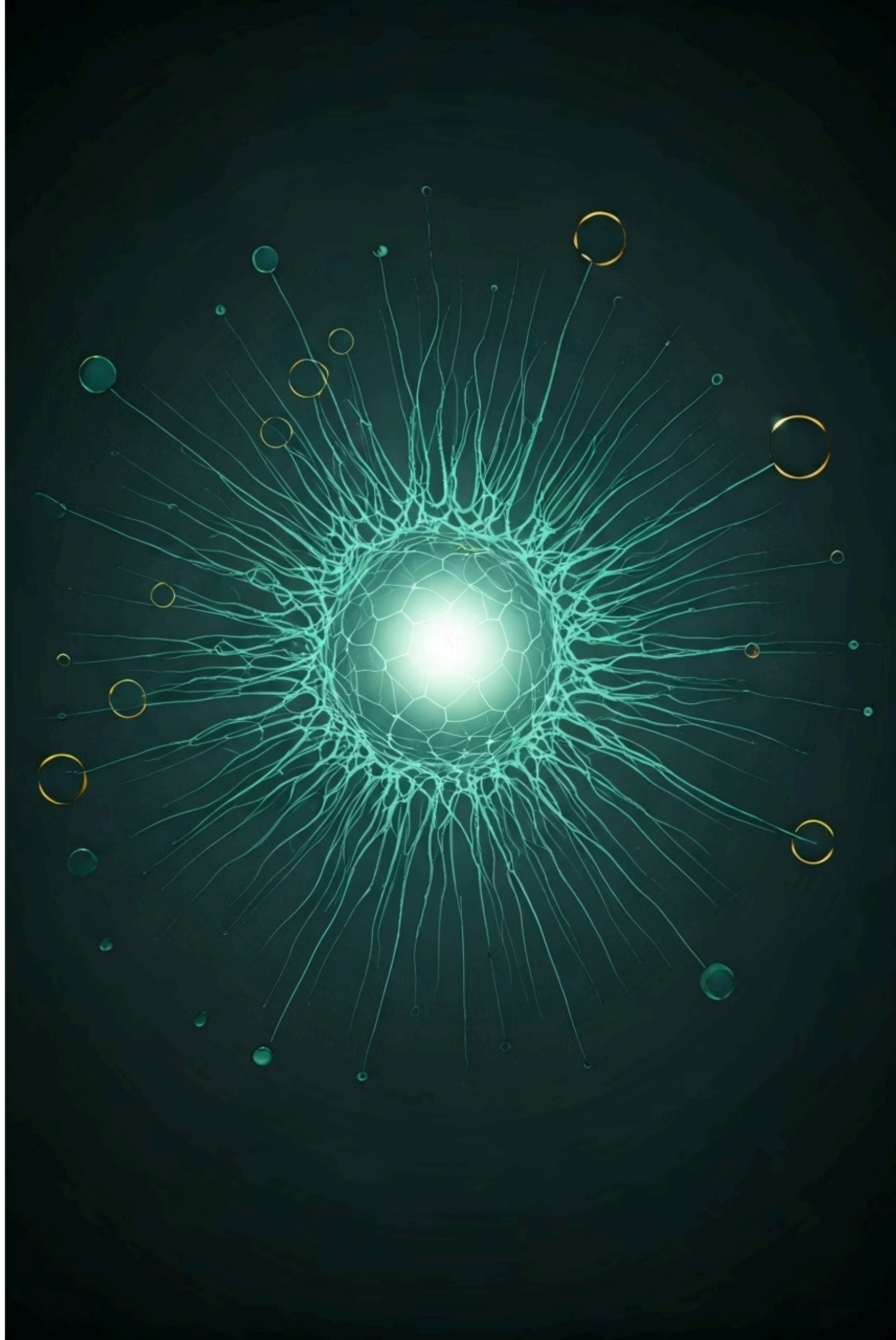
Introduction to LLMs & Tokenization

Welcome to our exploration of Large Language Models and tokenization fundamentals. We'll uncover how these powerful AI systems understand language.

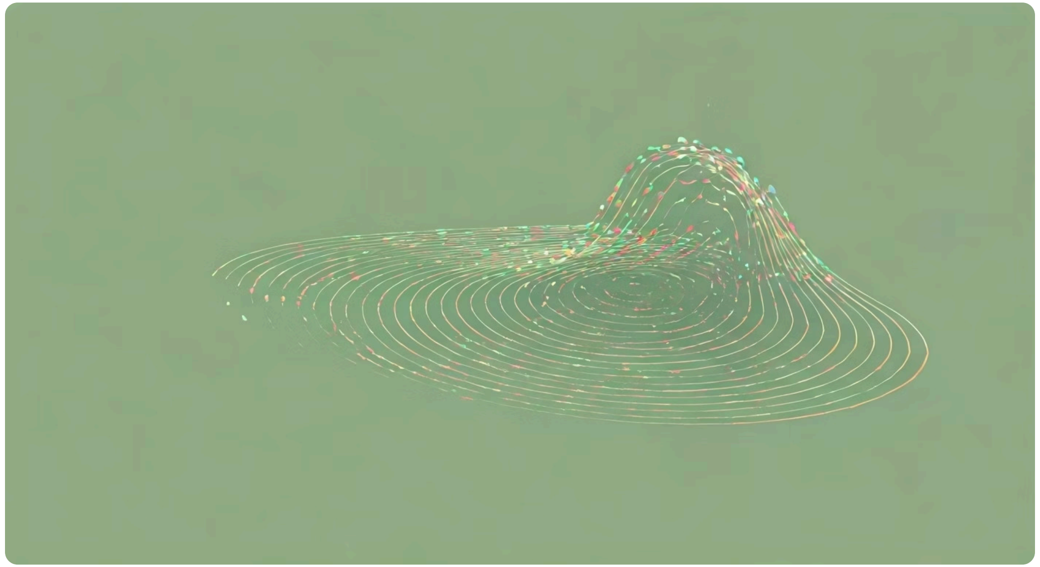
T **by The XYZ Company**

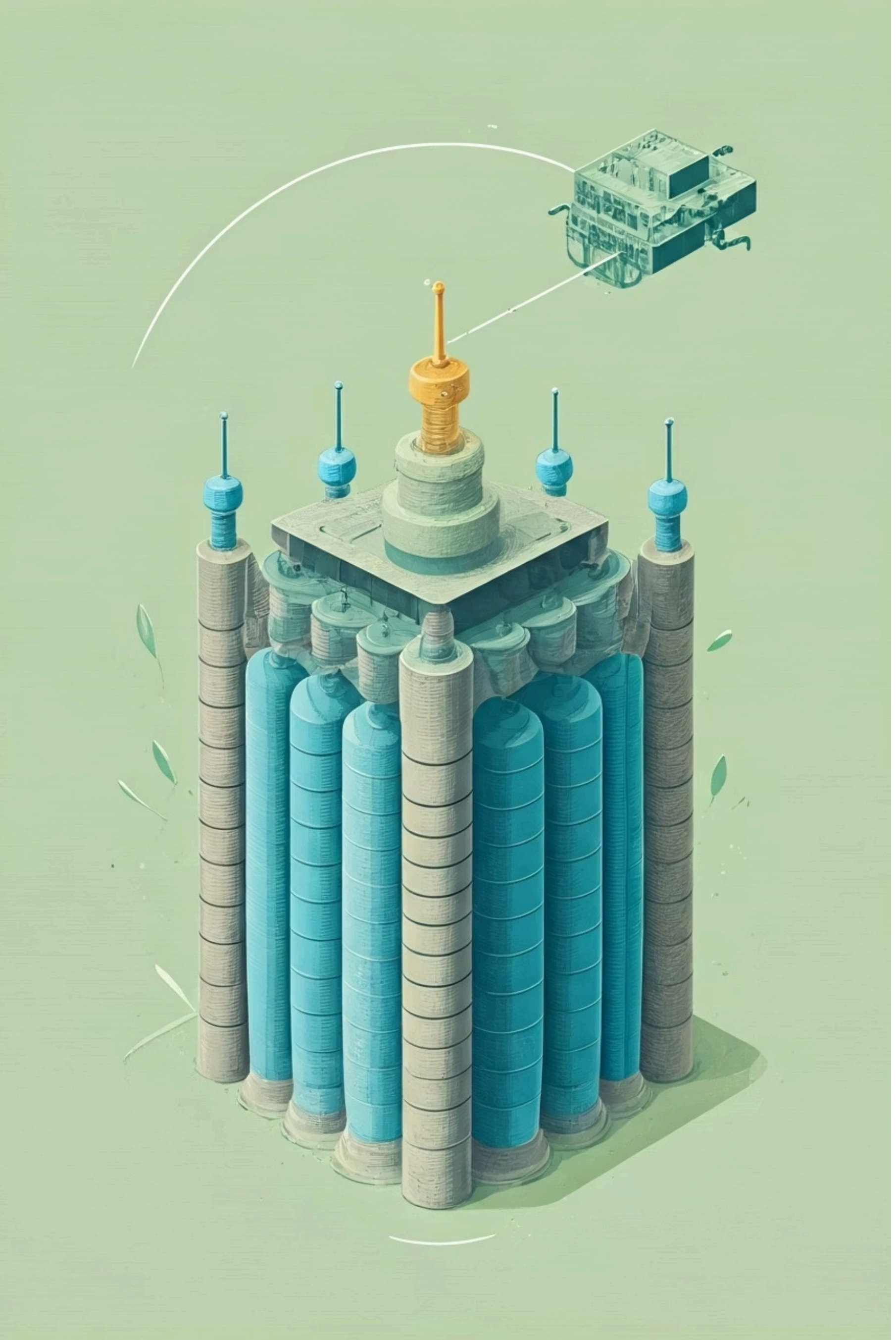


What Are Large Language Models (LLMs)?

LLMs are deep learning models trained on vast text corpora. They learn patterns and relationships in language without explicit programming.

These models can generate coherent text, translate languages, and summarize content with remarkable fluency.





Common LLM Architectures: The Transformer

Self-Attention

Captures relationships between words regardless of their distance in text.

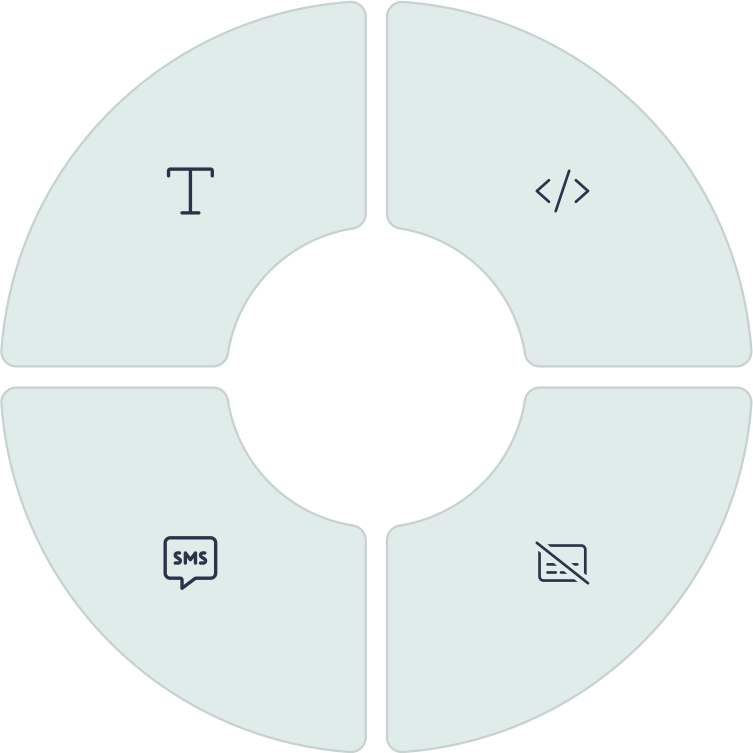
Layer Stacking

Multiple attention and feedforward layers process information hierarchically.

Parallel Processing

Enables efficient training and inference on massive datasets.

Capabilities of LLMs



Text Generation

Create human-like content for various purposes.

Coding

Write and debug programs in multiple languages.

Conversation

Engage in dialogue across diverse topics.

Translation

Convert text between hundreds of languages.

The Need for Subword Tokenization

Word-level tokenization struggles with unlimited vocabulary. New or rare words become "unknown" tokens.

Subword tokenization breaks words into meaningful pieces. This creates a balance between vocabulary size and coverage.

> Example: "unhappiness" → ["un", "happy", "ness"]

Popular Subword Tokenizers

Byte-Pair Encoding (BPE)

Iteratively merges most frequent character pairs. Used in GPT models.

WordPiece

Splits words based on likelihood scores. Powers BERT and derivatives.

SentencePiece

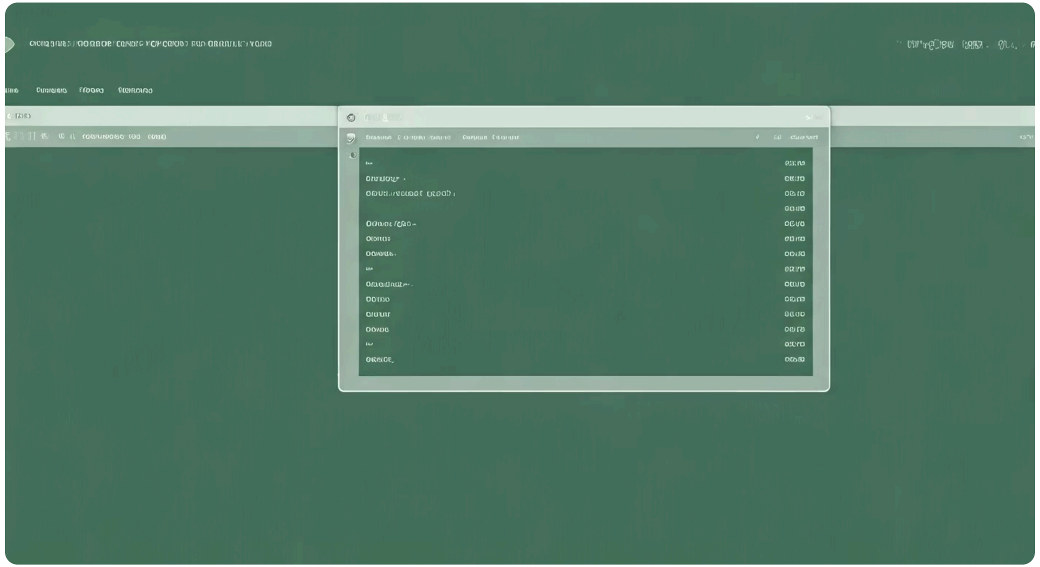
Language-agnostic approach. Treats text as Unicode sequences.

Tokenization in Practice

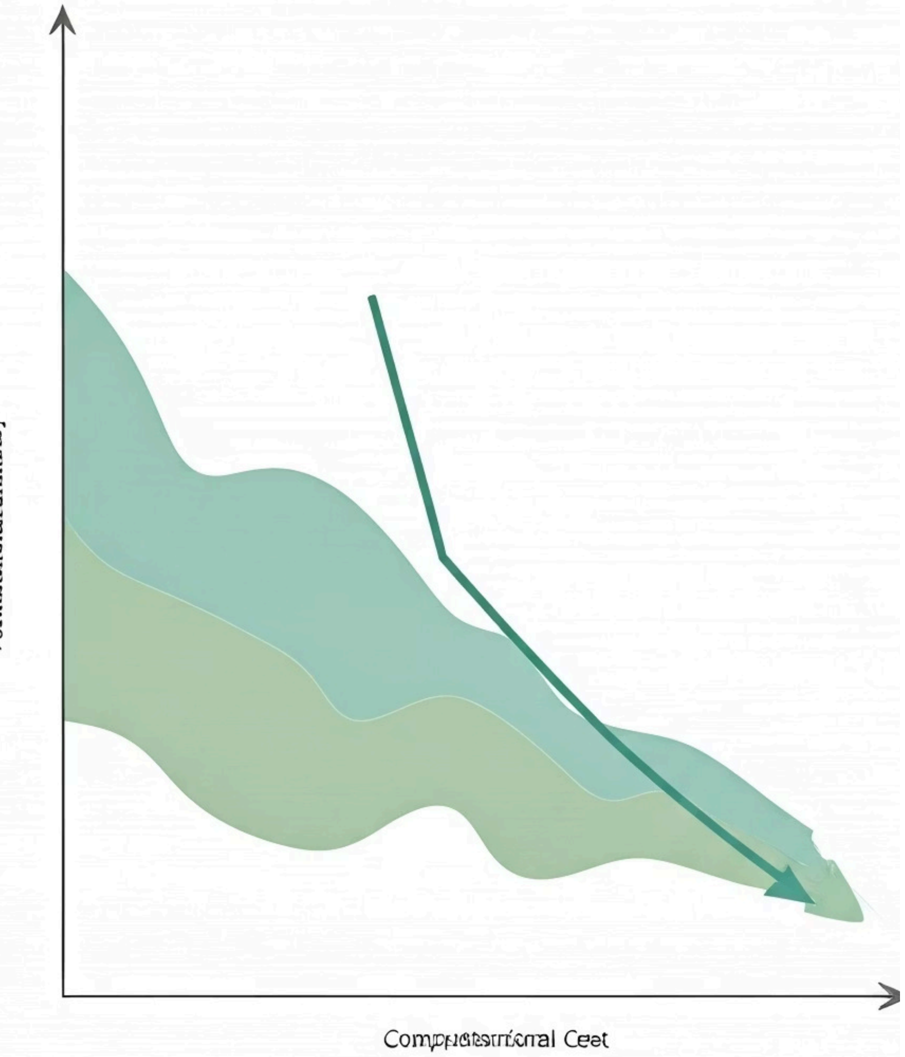
Using Hugging Face Tokenizers

```
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("gpt2")
tokens = tokenizer.tokenize("unhappiness")
print(tokens)  # ['un', 'happiness']
```



Tokenizers library offers fast, consistent implementations across model architectures.



Impact of Tokenization on Performance & Cost

Context Window

Token count limits how much text the model can process at once.

Computation

More tokens mean more operations and higher processing costs.

Efficiency

Better tokenizers compress text into fewer tokens, reducing costs.