**TRANSFORMERS**

**A diagram of a transformer

AI-generated content may be incorrect.**

**What is a Transformer?**

A **transformer** is a **type of model architecture** used in machine learning, especially in natural language processing (NLP). It's the architecture behind models like **ChatGPT**, **BERT**, and **GPT-4**.

Transformers revolutionized the way machines understand and generate human language. They replaced older models like RNNs and LSTMs because they're **faster**, **more accurate**, and **more scalable**.

**Core Idea**

At a high level, a **transformer** takes a sequence of data (like a sentence) and processes all the parts **at the same time** to figure out what each part means, especially in the context of the rest.

For example, in the sentence:

“The cat sat on the mat.”

A transformer can understand that:

* "cat" is the subject,
* "sat" is the action,
* "on the mat" is where the action happened,  
  …all at once.

**Key Components of a Transformer**

There are **two main parts** in a transformer:

**1. Encoder**

* Takes the input (e.g., an English sentence).
* Understands and converts it into a format the computer can work with (embeddings + context).

**2. Decoder**

* Takes the encoded input and turns it into the desired output (e.g., a French translation).

In models like GPT, we often **only use the decoder** to generate text, while in translation tasks like in Google Translate, we use **both encoder and decoder**.

A diagram of a transformer architecture

AI-generated content may be incorrect.

**Working Of The Transformer**

Let’s dive deeper into how a transformer works **step by step**:

**Step 1: Input Embeddings**

Words are not numbers, but computers need numbers to work. So we turn each word into a **vector** (a list of numbers) using something called **word embeddings**.

For example:

* "cat" → [0.1, 0.3, -0.2, ...]
* "sat" → [0.2, -0.1, 0.4, ...]

This vector represents the **meaning** of the word in some way.

**Step 2: Positional Encoding**

Transformers process the whole sentence at once, **not word-by-word**, so they need a way to know the **order** of words.  
They add something called **positional encoding** to each word vector, so the model knows, for example, that "The" comes before "cat".

Think of it like saying:

* “This is word #1, and here’s its meaning.”
* “This is word #2, and here’s its meaning.”

**Step 3: Self-Attention (The Magic)**

This is the most important part.

**Self-attention** allows the model to look at **all other words** in the sentence to understand the meaning of each word **in context**.

**Example:**

In the sentence:

“She gave the book to her friend because **she** loved reading.”

What does "**she**" refer to? The model uses self-attention to figure that out by comparing "**she**" to "her friend" and "she" earlier in the sentence.

The idea is: for each word, figure out **how much attention to pay** to every other word.

**Step 4: Feed Forward Network**

After attention, each word’s data (now updated with context) goes through a small neural network to process it further.

**Step 5: Stacking Layers**

The process above (attention + feed-forward) is repeated **multiple times** in layers. Each layer refines the understanding of each word based on more complex patterns.

**Step 6: Output**

In the encoder-decoder setup:

* The **encoder** gives the context-rich information to the decoder.
* The **decoder** uses this to generate output, **one word at a time**, like writing a sentence.

In GPT-like models, the model generates the next word based on previous words **using only the decoder** (called causal or autoregressive decoding).

A diagram of a book

AI-generated content may be incorrect.

**Why Transformers Work So Well**

1. **Parallel Processing**: Unlike RNNs, which process one word at a time, transformers handle **entire sequences at once**—much faster!
2. **Contextual Understanding**: Through **attention**, each word is understood in the context of **all other words**.
3. **Scalability**: Transformers scale very well with more data and computing power.

**Examples of Transformer-Based Models**

A screenshot of a cell phone

AI-generated content may be incorrect.

One of the most well-known transformer models is **GPT** (Generative Pretrained Transformer), developed by OpenAI. It started with GPT-2, which was already impressive at generating human-like text, and evolved into GPT-3 and **GPT-4**, which power many conversational AI tools, including **ChatGPT**. The latest version, **GPT-4o**, is even more advanced and can handle not just text, but also **images and audio**—making it a **multimodal** model.

For tasks that require **understanding language** rather than generating it, models like **BERT** (Bidirectional Encoder Representations from Transformers) are very popular. BERT was created by Google and is great at things like **text classification**, **question answering**, and **sentiment analysis**. There are several improvements on BERT too—like **RoBERTa** (a robust version by Facebook), **ALBERT** (a lighter, faster version), and **DistilBERT** (a compact version designed to be more efficient while keeping most of BERT’s accuracy).

Other models are built for **translation**, **summarization**, and handling multiple tasks. For example, **T5** (Text-to-Text Transfer Transformer) treats every task as a form of text generation—even things like classification are done by producing a label as text. T5 has a multilingual version called **mT5**. Another popular model is **BART**, which blends ideas from both BERT and GPT, and is often used for summarizing long documents or translating between languages. **MarianMT** is another transformer model specialized in machine translation, and **Meta’s NLLB (No Language Left Behind)** is designed to translate between 200+ languages, many of which are low-resource.

There are also **multimodal transformers**—models that understand both **language and images** (and sometimes even audio). For instance, **CLIP** by OpenAI connects images and text, allowing systems to "see" and "read" at the same time. **DALL·E** uses transformers to generate images from text prompts. **Flamingo**, by DeepMind, is a multimodal model trained to understand sequences of text and images together. The most advanced example is **GPT-4o**, which can accept text, image, and audio input and generate responses across all those types.