The Transformer architecture is a deep learning model built around self-attention, enabling parallel processing of sequences and capturing long-range dependencies. It consists of an encoder–decoder design with layers of attention, feed-forward networks, normalization, and residual connections GeeksForGeeks DataCamp Jay Alammar.

🧩 **Core Concepts of Transformer Architecture**

1. Encoder–Decoder Structure

- **Encoder**: Processes input sequences into contextual representations.
- **Decoder**: Generates output sequences using encoder outputs and previously generated tokens.
- This design is especially effective for tasks like **machine translation** and **text generation** GeeksForGeeks.

---

2. Self-Attention Mechanism

- **Purpose**: Allows each token to "attend" to other tokens in the sequence.
- **Computation**:
  - Input tokens are projected into **Query (Q), Key (K), and Value (V)** vectors.
  - Attention scores are calculated as:
    [\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V]
- **Benefit**: Captures relationships regardless of distance in the sequence DataCamp.

---

3. Multi-Head Attention

- Multiple attention "heads" run in parallel.
- Each head learns different aspects of relationships (syntax, semantics, dependencies).
- Outputs are concatenated and linearly transformed.

---

4. Positional Encoding

- Since transformers don't have recurrence, positional encodings inject sequence order.
- Commonly sinusoidal functions or learned embeddings.

---

5. Feed-Forward Networks

- After attention, each token passes through a fully connected feed-forward network.
- Adds nonlinearity and richer representation.

---

6. Residual Connections & Normalization

- **Residuals**: Skip connections help gradient flow and stabilize training.
- **LayerNorm**: Normalization ensures stable activations.

🔍 **Advanced Features**

- **Parallelization**: Unlike RNNs, transformers process sequences in parallel → faster training.
- **Scalability**: Forms the backbone of LLMs like GPT, BERT, and LaMDA DataCamp.
- **Variants**:
  - **Encoder-only** (BERT) → great for classification.
  - **Decoder-only** (GPT) → great for generation.
  - **Encoder–decoder** (T5) → great for translation and summarization.
- **Improvements**: Multi-query attention, rotary embeddings (RoPE), and sparse attention for efficiency Jay Alammar.

📊 **Enterprise AI Context**

For your **enterprise AI prep**:

- Transformers are the **foundation of LLMs** used in RAG pipelines.
- Understanding attention and encoder/decoder roles helps explain **why LLMs can be adapted with fine-tuning or retrieval**.
- In interviews, emphasize how **transformers enable scalability and contextual reasoning**, which is critical for enterprise copilots like Salesforce Agentforce.

In short: Transformers revolutionized NLP by replacing recurrence with self-attention, enabling parallelism, scalability, and deep contextual understanding. They are the backbone of modern LLMs and enterprise AI systems.

Sources: GeeksForGeeks DataCamp Jay Alammar