GPT

✅ GPT – Generative Pre trained Transformer

🔷 Developed by **OpenAI** (GPT-1, GPT-2, GPT-3, GPT-4)
🔷 It's an **autoregressive language model**
🔷 Learns to predict the **next token** in a sequence, left to right.

1. Architecture

Feature	Description
Model Type	Decoder-only Transformer
Self-Attention	**Causal (masked)** self-attention – can't see future tokens
Directionality	**Unidirectional** (left-to-right)
Layers	Multiple Transformer decoder blocks with self-attention and feed-forward layers
Embedding	Token embeddings + positional embeddings

2. Tokenization

- Uses **Byte Pair Encoding (BPE)** for tokenization
- Efficiently breaks words into subword units
- Helps handle unknown and rare words

3. Training Objective

- **Autoregressive Language Modeling**
- Given tokens t1, t2, ..., tn, predict the next token tn+1
- Objective: maximize the likelihood of the next token

4. Text Generation Techniques

Technique	Description
Greedy Search	Always picks the most likely next word
Beam Search	Keeps multiple sequences (beams) and selects the best
Sampling	Randomly samples based on probabilities
Top-k Sampling	Samples from top k most likely tokens
Top-p Sampling	Samples from the smallest set of tokens whose cumulative probability ≥ p (nucleus sampling)

5. GPT Learning Modes

Mode	Explanation
Zero-shot	Model answers without any examples
Few-shot	Model sees a few examples in the prompt
In-context Learning	Learns patterns during inference just by looking at context examples

6. Applications

- Chatbots (e.g., ChatGPT)
- Code completion (Codex)
- Content generation (blogs, tweets)
- Question answering
- Summarization (limited)
- Creative writing

7. Limitations

- **No bidirectional context** → can't understand the full sentence like BERT
- May generate **incoherent or biased** outputs
- Requires a lot of **compute for training**
- Lacks **true reasoning** (just next-token prediction)

8. GPT vs BERT (Quick Comparison)

Feature	GPT	BERT
Type	Decoder-only	Encoder-only
Direction	Left-to-right	Bidirectional
Use Case	Generation	Understanding
Task Format	Text generation	Classification, QA
Pretraining	Predict next token	Masked Language Model (MLM)