BERT

✅ BERT: Complete Revision Notes

🔷 1. What is BERT?

- **BERT = Bidirectional Encoder Representations from Transformers**
- Developed by **Google (2018)**.
- Based on the **Transformer Encoder** (only the encoder side is used).
- Allows **contextual understanding of words in both directions** (left-to-right & right-to-left simultaneously).

🔷 2. Why BERT?

- Traditional NLP models (like Word2Vec, GloVe) are **context-free**.
  - *Example:* "bank" in "river bank" vs "money bank" – same embedding.
- BERT provides **contextual embeddings**: "bank" has different vectors depending on context.

🔷 3. Architecture of BERT

- Based on the **Transformer Encoder stack**.
- Common versions:
  - bert-base-uncased: 12 layers (transformer blocks), 768 hidden units, 12 attention heads, 110M parameters.
  - bert-large-uncased: 24 layers, 1024 hidden units, 16 heads, 340M parameters.
- **Input Embedding = Token Embedding + Segment Embedding + Positional Embedding**

🔷 4. BERT Input Format

BERT expects a specific input format:

[CLS] Sentence A [SEP] Sentence B [SEP]

- **[CLS]**: Special classification token used for tasks like classification.
- **[SEP]**: Separator token between two sentences.

🔷 5. BERT Tokenization (WordPiece)

- Words are broken into **subword units**.
- Example:
  - "playing" → "play", "##ing"
- Helps handle **unknown and rare words**.

🔷 6. BERT Pretraining Tasks

📌 a. Masked Language Modeling (MLM)

- Randomly masks 15% of input tokens.
- The model predicts these masked tokens.

- Unlike traditional LMs, it uses **bidirectional context**.

📌 **b. Next Sentence Prediction (NSP)**

- Given two sentences, predict if **Sentence B follows Sentence A** in original text.
- Input:
  - Sentence A → "The man went to the store."
  - Sentence B → "He bought a gallon of milk." → **Label: IsNext**
  - Negative example → random sentence → **Label: NotNext**

🔷 7. Fine-Tuning BERT

- BERT can be fine-tuned for downstream tasks with minimal changes:
  - **Text Classification** (add classifier head to [CLS] token)
  - **Named Entity Recognition (NER)** (token-level classification)
  - **Question Answering** (start & end token prediction)
  - **Text Similarity / Semantic Search**
  - **Summarization / Paraphrasing** (with encoder-decoder variants)

🔷 8. Applications of BERT

- Sentiment Analysis
- Intent Classification
- Question Answering (like SQuAD)
- Semantic Textual Similarity
- Sentence Embedding Generation
- Named Entity Recognition (NER)
- Zero-shot Learning (via models like BART or T5)

🔷 9. Popular BERT Variants

Variant	Purpose
DistilBERT	Smaller, faster version of BERT (student model)
RoBERTa	Robustly optimized BERT, no NSP, trained longer
ALBERT	A Lite BERT – parameter reduction for speed
TinyBERT	Very small, efficient model for mobile devices
BioBERT, SciBERT	Domain-specific (biomedical, scientific)
mBERT	Multilingual BERT (100+ languages)

🔷 10. Limitations of BERT

- **No decoder**, hence not good for **generative tasks** (e.g., text

generation, summarization).

- **Heavy computation** and **memory usage**.
- Trained on relatively **short input length** (up to 512 tokens).

🔷 11. Key Differences with GPT & T5

Model	Direction	Architecture	Pretraining Task	Use Case
BERT	Bidirectional	Encoder-only	MLM + NSP	Classification, QA
GPT	Unidirectional (Left-to-right)	Decoder-only	Causal LM	Text generation
T5	Encoder-Decoder	Seq2Seq	Text-to-Text	Translation, Summarization