**Short Answers**

**1Ans :**

1. **Stemming vs. Lemmatization**:
   * Stemming: Cuts words to rough roots (e.g., "running" → "run").
   * Lemmatization: Converts words to proper dictionary forms (e.g., "running" → "run", but "better" → "good").
2. **Stopwords Removal**:
   * **Good**: Removes useless words (like "the", "and") to focus on important words. Helps in search/text analysis.
   * **Bad**: Can remove important words in short texts or phrases (e.g., "not happy" → "happy" changes meaning).

**2Ans :**

1. **NER vs POS Tagging:**
   * NER identifies real-world objects (people, places, dates) and categorizes them.
   * POS tagging marks each word's grammatical role (noun, verb, adjective).

Example: In "Apple launched a phone":

* + NER: "Apple" → ORG (organization)
  + POS: "Apple" → PROPN (proper noun), "launched" → VERB

1. **Real-world NER Applications:**
   * Customer Support: Auto-tagging tickets with company/product names to route them.
   * Healthcare: Extracting patient info (medications, conditions) from medical records.

**3Ans :**

1. **Why divide by √d?**
   * To prevent the dot products from becoming too large when the key dimension (d) is big.
   * Large values make softmax output very sharp (close to 0 or 1), which can slow down learning.
   * Scaling keeps values at a reasonable size for stable gradients.
2. **How self-attention helps understand word relationships:**
   * It lets each word "look" at all other words in the sentence.
   * Computes how much attention to pay to each word when processing another word.
   * For example, for the word "bank", it can decide whether to focus more on "money" or "river" nearby words.

**4 Ans :**

1. **BERT vs GPT architecture difference:**
   * BERT uses an encoder architecture (bi-directional - sees full context)
   * GPT uses a decoder architecture (uni-directional - predicts next word)
   * Example: BERT can understand word context from both sides, while GPT generates text sequentially
2. **Benefits of pre-trained models:**
   * Saves time/resources (no need to train from scratch)
   * Already learned rich language patterns from massive datasets
   * Can be fine-tuned for specific tasks with small datasets
   * Better performance than training from scratch on limited data
   * Example: BERT already understands grammar, word meanings, and relationships before fine-tuning