Short Answers

Poorna Chander Oruganti

70771980

1. **What is the difference between stemming and lemmatization? Provide examples with the word "running."**

   Stemming is a crude, rule-based chopping of word endings. It may produce non-words:

   - *running* → stemmer → **"runn"** (or "run" with some stemmers)

   Lemmatization uses vocabulary and part-of-speech to return a real base form:

   - ***running*** **→ lemmatizer → "run"**

2. **Why might removing stop words be useful in some NLP tasks, and when might it actually be harmful?**

   Helpful

   - Cuts down vocabulary size and noise in bag-of-words or topic-modeling tasks
   - Speeds up training and often boosts accuracy in basic text-classification

   Harmful

   - Removes crucial words like negations ("not," "never") that flip sentiment
   - Strips out function words needed for tasks that rely on syntax or nuance (e.g. **machine translation, question answering)**

3. **How does NER differ from POS tagging in NLP?**

   POS tagging labels *every* token with its grammatical role (e.g. "bank" → noun, "runs" → verb).

   NER selectively recognizes and classifies *named* items—people, organizations, locations, dates, etc.—grouping one or more tokens into semantic entities ("Barack Obama" → PERSON, "United Nations" → ORG)

4. **Describe two applications that use NER in the real world (e.g., financial news, search engines).**

   Financial news analytics: Automatically identify company names, ticker symbols, and financial events in earnings reports or press releases to power trading signals or risk-monitoring dashboards.

   Search engines / QA systems: Detect entities in user queries (e.g., "hotels in Paris" → GPE=Paris) so the engine can return targeted results or feed downstream question-answering modules with precise semantic slots.

5. **Why do we divide the attention score by √d in the scaled dot-product attention formula?**

   When you take the dot-product of Q and K, the raw scores grow with the key dimension d, which makes the softmax either extremely flat or sharply peaked. Dividing by √d normalizes those scores to a stable range, so gradients stay well-behaved during training and the model can learn more reliably.

6. **How does self-attention help the model understand relationships between words in a sentence?**

   Self-attention lets each word's representation "look at" every other word in the same sentence. By computing a weighted sum of all token embeddings (where the weights reflect pairwise similarity), the model dynamically encodes context—so "bank" in "river bank" attends more to "river," whereas in "bank account" it attends more to "account," capturing meaning via those learned connections.

7. **What is the main architectural difference between BERT and GPT? Which uses an encoder and which uses a decoder?**

   BERT is an **encoder-only**, bidirectional Transformer: every token attends to both left and right context simultaneously.

   GPT is a **decoder-only**, causal (left-to-right) Transformer: each token attends only to earlier positions, making it autoregressive for text generation.

8. **Explain why using pre-trained models (like BERT or GPT) is beneficial for NLP applications instead of training from scratch.**

   Data Efficiency: You leverage huge-scale language knowledge learned on unlabeled corpora, so downstream tasks need far less labeled data.

   Faster Training: You fine-tune a pre-trained backbone in hours or days instead of training from scratch for weeks.

   Better Generalization: Models come with rich, transferable linguistic and world knowledge, improving accuracy and robustness on diverse NLP tasks.