**Fundamentals & Definitions**

Q: What type of machine learning architecture are most modern LLMs based on?

A: Transformer architecture.

Q: What is the primary method of training for LLMs?

A: Self-supervised learning (specifically, masked language modeling or next-token prediction).

Q: What is the main benefit of "large" in LLMs?

A: The ability to learn more complex patterns, nuances, and a broader range of knowledge from vast datasets.

**Architecture & Training**

**Q: What are the two main components of the Transformer architecture?**

A: Encoder and Decoder. (Note: Many LLMs like GPT are decoder-only).

**Q: What is the purpose of "attention mechanisms" in Transformers?**

A: To allow the model to weigh the importance of different words in the input sequence when processing each word.

**Q: What is "self-attention"?**

A: A mechanism where the model attends to different parts of the same input sequence to understand relationships between words.

**Q: What is "pre-training" in the context of LLMs?**

A: The initial phase where the LLM learns general language patterns and knowledge from a massive, diverse text dataset.

**Q: What is "fine-tuning" an LLM?**

A: Further training an already pre-trained LLM on a smaller, task-specific dataset to adapt it for a particular application (e.g., sentiment analysis, summarization).

**Q: What are "parameters" in an LLM?**

A: The trainable weights and biases within the neural network that the model learns during training. The "size" of an LLM often refers to its number of parameters.

**Q: Why are GPUs crucial for LLM training?**

A: They excel at parallel processing, which is essential for the massive matrix multiplications involved in neural network computations.

**Capabilities & Applications**

**Q: Name three common applications of LLMs.**

A: Text generation (creative writing, emails), summarization, question answering, translation, code generation, chatbots. (Any three are fine).

**Q: What is "few-shot learning" for LLMs?**

A: The ability of a pre-trained LLM to perform a new task with only a small number of examples, often provided as part of the prompt.

**Q: What is "zero-shot learning" for LLMs?**

A: The ability of an LLM to perform a new task without any explicit examples, relying solely on its pre-trained knowledge and understanding of the instructions.

**Q: How can LLMs be used for text summarization?**

A: By prompting the model to condense a longer text into a shorter, coherent version while retaining key information.

**Technical Concepts**

**Q: What is "tokenization" in LLMs?**

A: The process of breaking down text into smaller units (tokens), which can be words, subwords, or characters, for the model to process.

**Q: What is an "embedding" in LLMs?**

A: A dense vector representation of a word, token, or phrase that captures its semantic meaning and relationships to other words.

**Q: What is "perplexity" as a metric for LLMs?**

A: A measure of how well a probability model predicts a sample. Lower perplexity generally indicates a better model.

**Q: What is the "context window" (or "context length") of an LLM?**

A: The maximum number of tokens an LLM can consider at one time when generating a response.

**Q: What is a "prompt" in the context of LLMs?**

A: The input text or instruction given to an LLM to guide its generation.

**Q: What is "prompt engineering"?**

A: The art and science of crafting effective prompts to elicit desired behaviors and outputs from LLMs.

**Q: What is "temperature" in LLM inference?**

A: A hyperparameter that controls the randomness or creativity of the LLM's output. Higher temperature leads to more varied (and potentially less coherent) results.

**Q: What is "Top-P sampling" (or "nucleus sampling")?**

A: A decoding strategy that samples from the smallest set of tokens whose cumulative probability exceeds a certain threshold 'P'.

**Advanced & Future Concepts**

**Q: What is the difference between an encoder-decoder Transformer and a decoder-only Transformer?**

A: Encoder-decoder models (like T5, BART) are good for sequence-to-sequence tasks (e.g., translation, summarization). Decoder-only models (like GPT) are primarily for generative tasks (e.g., text completion, chatbots).

**Q: What is "Multimodal LLM"?**

A: An LLM that can process and generate content across multiple modalities, such as text, images, audio, and video.

**Q: What is the role of "prompt chaining" in complex LLM tasks?**

A: Breaking down a complex task into smaller sub-tasks, with the output of one prompt becoming the input for the next, to guide the LLM through a multi-step process.

**Q: What is "in-context learning"?**

A: The ability of LLMs to learn new tasks or adapt to new styles by observing examples provided directly within the prompt, without requiring explicit fine-tuning.

**Q: What is "retrieval-augmented generation" (RAG)?**

A: A technique that combines an LLM with a retrieval system to pull relevant information from an external knowledge base, allowing the LLM to generate more accurate and up-to-date responses.

**Q: How do LLMs differ from traditional rule-based chatbots?**

A: LLMs learn from massive datasets and can generate novel and flexible responses, while rule-based chatbots rely on pre-defined scripts and patterns.

**Q: What is "quantization" in LLMs?**

A: A technique to reduce the precision of the numerical representations of model weights (e.g., from 32-bit to 8-bit), thereby reducing model size and improving inference speed, often with minimal performance loss.

**Q: What is the concept of an "LLM agent"?**

A: An LLM that can perceive its environment, plan actions, use tools, and execute those actions to achieve a goal, often involving multiple steps and interactions.