🧠 The Ultimate LLMs Interview Question Bank (50 Questions)

Core Concepts

1. What does tokenization entail, and why is it critical for LLMs?

Answer: Tokenization is the process of breaking text down into smaller pieces called "tokens," which can be words, parts of words, or even single characters. For instance, the word "artificial" might be split into "art," "ific," and "ial". This is critical because LLMs cannot read raw text; they need these tokens converted into numbers to process language. It helps the model handle different languages, manage rare words, and run more efficiently.

2. How does the attention mechanism function in transformer models?

Answer: The attention mechanism helps the model decide which words in a sentence are most important when trying to understand context. It's like highlighting keywords; in the sentence "The cat chased the mouse," attention helps the model link "mouse" to "chased" so it understands who was being chased. It works by calculating similarity scores between different tokens.

3. What is the context window in LLMs, and why does it matter?

Answer: The context window is essentially the model's short-term memory. It defines how much text (measured in tokens) the model can look at and process at one single time. A larger window (e.g., 32,000 tokens) allows the model to remember more of a conversation or document, leading to more coherent and relevant answers, though it costs more computing power.

4. What distinguishes LORA from QLORA in fine-tuning LLMs?

Answer: LORA (Low-Rank Adaptation) is a method to fine-tune models efficiently by only updating a small part of the model's layers rather than the whole thing. QLORA takes this a step further by compressing the model using "quantization" (e.g., using 4-bit precision instead of standard precision). This drastically reduces memory usage, allowing massive models to be fine-tuned on smaller, cheaper hardware.

5. How does beam search improve text generation compared to greedy decoding?

Answer: Greedy decoding always picks the single most likely next word, which can lead to repetitive or boring text. Beam search is smarter; it explores several different possible paths (sequences of words) at once. By keeping the top few best options (called "beams") at each step, it produces sentences that make more sense overall.

6. What role does temperature play in controlling LLM output?

Answer: Temperature controls how creative or random the model's responses are.

- **Low temperature (e.g., 0.3):** The model plays it safe, choosing only the most likely words. The output is predictable and focused.
- **High temperature (e.g., 1.5):** The model takes risks, choosing less likely words. The output is more creative but can be nonsensical.

7. What is masked language modeling, and how does it aid pretraining?

Answer: Masked language modeling (MLM) is a training game where you hide random words in a sentence and ask the model to guess them based on the context. For example, "The [MASK] sat on the mat." This forces the model to understand the relationship between words in both directions (before and after the blank), improving its overall understanding of language.

8. What are sequence-to-sequence models, and where are they applied?

Answer: Sequence-to-sequence (Seq2Seq) models are designed to take one sequence of text and turn it into another. They use an "encoder" to understand the input and a "decoder" to create the output. They are perfect for tasks where the input and output lengths change, like translating English to Spanish or summarizing a long article.

9. How do autoregressive and masked models differ in LLM training? Answer:

- **Autoregressive models (like GPT):** Read text from left to right and try to predict the *next* word. They are great at writing and generating text.
- **Masked models (like BERT):** Look at the whole sentence at once to guess missing words. They are better at *understanding* text for tasks like classification or sentiment analysis.

10. What are embeddings, and how are they initialized in LLMs?

Answer: Embeddings are lists of numbers (vectors) that represent words. Words with similar meanings have similar numbers. They start as random numbers or are borrowed from other pre-trained models. As the LLM trains, these numbers are adjusted so the model learns that "dog" is closer to "puppy" than to "car".

Training and Techniques

11. What is next sentence prediction, and how does it enhance LLMs?

Answer: Next Sentence Prediction (NSP) teaches a model to guess if two sentences logically follow one another. During training, the model looks at pairs of sentences and decides if they are connected or random. This helps the model understand flow and logic, which is essential for chatting or summarizing text.

12. How do top-k and top-p sampling differ in text generation? Answer: These are methods to pick the next word.

- **Top-k:** The model only considers the top "k" (e.g., 20) most likely words and ignores the rest.
- **Top-p (Nucleus):** The model considers the smallest group of words whose probabilities add up to "p" (e.g., 95%). Top-p is generally more flexible and produces more natural, varied text.

13. Why is prompt engineering crucial for LLM performance?

Answer: Prompt engineering is the art of writing clear instructions for the model. A specific prompt like "Summarize this in 100 words" works much better than a vague one. Good prompts help the model perform tasks it hasn't been specifically trained for (zero-shot learning) without needing to change the model itself.

14. How can LLMs avoid catastrophic forgetting during fine-tuning?

Answer: Catastrophic forgetting is when a model learns a new task but forgets what it learned before. To stop this, developers use techniques like:

- **Rehearsal:** Mixing old data in with the new training data.
- **Freezing:** Keeping important parts of the model unchanged while only training new parts.

15. What is model distillation, and how does it benefit LLMs?

Answer: Distillation is like a teacher (big model) teaching a student (small model). The small model learns to mimic the big model's answers. This creates a model that is much smaller and faster but almost as smart, making it possible to run AI on phones or smaller computers.

16. How do LLMs manage out-of-vocabulary (OOV) words?

Answer: If an LLM sees a word it doesn't know, it uses "subword tokenization". It breaks the unknown word into parts it *does* know. For example, it might split "cryptocurrency" into "crypto" and "currency". This allows it to understand and read words it has never seen before.

17. How do transformers improve on traditional Seq2Seq models?

Answer: Old models (RNNs) processed words one by one, which was slow. Transformers process entire sentences at once (parallel processing). They also use "attention" to better understand connections between words that are far apart in a sentence, making them faster and smarter.

18. What is overfitting, and how can it be mitigated in LLMs?

Answer: Overfitting happens when a model memorizes the training data instead of learning general rules, so it fails on new data. To fix this, developers use "dropout" (randomly turning off parts of the model during training) or "early stopping" (stopping training before the model starts memorizing).

19. What are generative versus discriminative models in NLP? Answer:

- **Generative models (like GPT):** Create new data. You give them a prompt, and they write a story or an answer.
- **Discriminative models (like BERT):** Classify data. You give them text, and they tell you if it's positive/negative or spam/not spam.

20. How does GPT-4 differ from GPT-3 in features and applications? Answer: GPT-4 is a significant upgrade. It can understand images as well as text (multimodal) , it has a much larger memory (context window) allowing it to read longer documents , and it makes fewer factual mistakes due to better training.

Architecture and Math

21. What are positional encodings, and why are they used?

Answer: Since transformers read all words at once, they don't inherently know which word comes first or last. Positional encodings are mathematical tags added to each word so the model understands the order, ensuring it knows the difference between "The dog bit the man" and "The man bit the dog".

22. What is multi-head attention, and how does it enhance LLMs?

Answer: Instead of focusing on the sentence in just one way, multi-head attention lets the model look at the sentence from multiple perspectives at the same time. One "head" might focus on grammar, while another focuses on meaning. This gives the model a deeper, more complex understanding.

23. How is the softmax function applied in attention mechanisms?

Answer: Softmax is a math function that turns raw scores into probabilities that add up to 100% (or 1). In attention, it helps the model decide exactly how much focus to put on each word. It highlights the most important words and ignores the irrelevant ones.

24. How does the dot product contribute to self-attention?

Answer: The dot product is a math operation used to calculate similarity. The model compares words (represented as numbers) using the dot product to see how related they are. A high score means the words are closely related and the model should pay attention to the connection.

25. Why is cross-entropy loss used in language modeling?

Answer: Cross-entropy loss is a way to measure how wrong the model's guess was. If the model predicts the wrong word, this score goes up. During training, the goal is to make this score as low as possible, teaching the model to make accurate predictions.

26. How are gradients computed for embeddings in LLMs?

Answer: Gradients are calculated using "backpropagation". The model looks at its error (loss) and works backward to see which embedding numbers contributed to the mistake. It then adjusts those numbers slightly to reduce the error for next time.

27. What is the Jacobian matrix's role in transformer backpropagation?

Answer: The Jacobian matrix helps handle the complex math of multivariable calculus needed for training. Since transformers have massive outputs with many dimensions, this matrix helps calculate exactly how to update the model's weights to fix errors.

28. How do eigenvalues and eigenvectors relate to dimensionality reduction?

Answer: These are concepts used to simplify data. They help identify the most important patterns (principal directions) in a massive dataset. By keeping only the most important patterns (high eigenvalues), you can simplify the data for the LLM without losing important information.

29. What is KL divergence, and how is it used in LLMs?

Answer: KL divergence measures how different two probability distributions are. In LLMs, it checks how far the model's predictions are from the "true" or desired distribution. It is often used during fine-tuning to make sure the model aligns with human preferences.

30. What is the derivative of the ReLU function, and why is it significant? Answer: ReLU is a function used to activate neurons in the model. Its derivative is simple: it's 1 if the input is positive, and 0 if negative. This simplicity makes it very fast to calculate and helps prevent math errors (vanishing gradients) that can break deep learning models.

31. How does the chain rule apply to gradient descent in LLMs?

Answer: The chain rule is a calculus rule for derivatives. In LLMs, it allows the model to calculate how an error at the very end of the process relates to a setting deep inside the model. It links the output error back through every layer to update the weights correctly.

32. How are attention scores calculated in transformers?

Answer: Attention scores are calculated by comparing a "Query" (what you are looking for) with a "Key" (what information is available) using a dot product. These scores are then normalized (using Softmax) to determine how much "Value" (information) to take from each word.

33. How does Gemini optimize multimodal LLM training?

Answer: Gemini is built from the ground up to handle both text and images together. It uses a unified structure so it doesn't need separate, heavy models for each type of data. It also uses smart data techniques to learn more efficiently from less labeled data.

Advanced Applications & Challenges

34. What types of foundation models exist?

Answer: Foundation models are the big, general-purpose models. They include:

- **Language Models:** Like GPT-4 (for text).
- **Vision Models:** Like ResNet (for images).
- **Generative Models:** Like DALL-E (for creating art).
- **Multimodal Models:** Like CLIP (combining text and images).

35. How does PEFT mitigate catastrophic forgetting?

Answer: Parameter-Efficient Fine-Tuning (PEFT) freezes most of the original model so it doesn't "forget" its basic knowledge. It only trains a tiny layer of new parameters to learn the new task. This keeps the model versatile while adapting it to new jobs.

36. What are the steps in Retrieval-Augmented Generation (RAG)?

Answer: RAG improves accuracy by letting the model "look up" information.

1. **Retrieval:** The system searches a database for documents related to your question.
2. **Ranking:** It sorts them to find the best ones.
3. **Generation:** It sends those documents to the LLM along with your question so the model can write an answer using facts it just read.

37. How does Mixture of Experts (MoE) enhance LLM scalability? Answer: Ideally, you don't need the whole brain for every question. MoE breaks the model into specialized sub-networks ("experts"). For any given question, a "gate" decides which experts are needed (e.g., only using 10% of the model). This makes massive models much faster and cheaper to run.

38. What is Chain-of-Thought (CoT) prompting, and how does it aid reasoning?

Answer: CoT is asking the model to "show its work" or think step-by-step. Instead of just asking for the answer to a math problem, you ask it to break down the steps. This helps the model solve complex logic puzzles where jumping straight to the answer often leads to mistakes.

39. How do discriminative and generative AI differ?

Answer: Discriminative AI is about making decisions or labels (e.g., "Is this a cat?"). Generative AI is about creation (e.g., "Draw a cat"). One analyzes existing data; the other creates new data.

40. How does knowledge graph integration improve LLMs?

Answer: A knowledge graph is a structured database of facts (like "Paris is the capital of France"). Connecting an LLM to this graph helps prevent hallucinations because the model can verify facts against the graph rather than just guessing based on probability.

41. What is zero-shot learning, and how do LLMs implement it?

Answer: Zero-shot learning is when an LLM does a task it was never specifically trained to do. Because it has read so much broad text, you can ask it to "Classify this movie review" and it understands the concept of classification and sentiment even without specific training examples.

42. How does Adaptive Softmax optimize LLMs?

Answer: Calculating probabilities for a massive vocabulary is slow. Adaptive Softmax speeds this up by grouping words. It spends less computational power on rare words that hardly ever appear, making the training process much faster and more efficient.

43. How do transformers address the vanishing gradient problem?

Answer: In deep networks, the training signal can fade away (vanish) before it reaches the early layers. Transformers use "Residual Connections" (shortcuts that let the signal skip layers) and "Layer Normalization" to keep the signal strong and stable throughout the entire model.

44. What is few-shot learning, and what are its benefits?

Answer: Few-shot learning is showing the model just a few examples (e.g., 3 or 4) of what you want it to do. It bridges the gap between zero-shot (no examples) and full training (thousands of examples). It’s cheap, fast, and very effective for specialized tasks.

45. How would you fix an LLM generating biased or incorrect outputs? Answer:

1. **Analyze:** Find out where the bias is coming from (is it the training data?).
2. **Enhance Data:** Clean the dataset to be more balanced and fair.
3. **Fine-Tune:** Retrain the model specifically to recognize and avoid those biases (using techniques like RLHF).

46. How do encoders and decoders differ in transformers?

Answer: The **Encoder** reads and understands the input text, turning it into abstract concepts. The **Decoder** takes those concepts and generates new text, one word at a time. For translation, the encoder reads the English, and the decoder writes the Spanish.

47. How do LLMs differ from traditional statistical language models?

Answer: Traditional models (like N-grams) just counted how often words appeared next to each other. LLMs use deep neural networks (Transformers) to understand complex context, long-term relationships, and meanings, allowing them to do diverse tasks rather than just predict the next word based on the last two.

48. What is a hyperparameter, and why is it important?

Answer: Hyperparameters are the settings you choose *before* training starts (like learning rate or batch size). They control how the model learns. If you set them wrong (e.g., learning rate too high), the model might never learn anything at all. Tuning them is key to getting a good model.

49. What defines a Large Language Model (LLM)?

Answer: An LLM is defined by two things: massive scale (billions of parameters) and massive training data. This combination allows it to understand and generate human-like language generally, rather than being good at just one specific narrow task.

50. What challenges do LLMs face in deployment? Answer: Putting LLMs into the real world is hard because:

- **Cost:** They require expensive, powerful hardware.
- **Bias:** They can accidentally be racist or sexist based on bad training data.
- **Privacy:** They might accidentally reveal private data they were trained on.
- **Black Box:** It is hard to explain *why* they gave a specific answer.