



Prepare Smart for Success Free Oracle 1Z0-1127-25 Exam Questions and Answers

Ready to pass faster? Grab free and updated Oracle Cloud Infrastructure 2025 Generative AI Professional exam PDF questions now. Get authentic 1Z0-1127-25 dumps packed with verified answers and secure your certification success with [PrepBolt](https://prepbolt.com/1Z0-1127-25.html) 1Z0-1127-25 exam pdf questions and answers.

Thank you for Downloading 1Z0-1127-25 exam PDF Demo

<https://prepbolt.com/1Z0-1127-25.html>

QUESTIONS & ANSWERS
DEMO VERSION
(LIMITED CONTENT)

Question 1

Question Type: MultipleChoice

What is prompt engineering in the context of Large Language Models (LLMs)?

Options:

- A- Iteratively refining the ask to elicit a desired response
- B- Adding more layers to the neural network
- C- Adjusting the hyperparameters of the model
- D- Training the model on a large dataset

Answer:

A

Explanation:

Comprehensive and Detailed In-Depth Explanation=

Prompt engineering involves crafting and refining input prompts to guide an LLM to produce desired outputs without altering its internal structure or parameters. It's an iterative process that leverages the model's pre-trained knowledge, making Option A correct. Option B is unrelated, as adding layers pertains to model architecture design, not prompting. Option C refers to hyperparameter tuning (e.g., temperature), not prompt engineering. Option D describes pretraining or fine-tuning, not prompt engineering.

: OCI 2025 Generative AI documentation likely covers prompt engineering in sections on model interaction or inference.

Question 2

Question Type: MultipleChoice

An AI development company is working on an AI-assisted chatbot for a customer, which happens to be an online retail company. The goal is to create an assistant that can best answer queries regarding the company policies as well as retain the chat history throughout a session. Considering the capabilities, which type of model would be the best?

Options:

- A- A keyword search-based AI that responds based on specific keywords identified in customer queries.
- B- An LLM enhanced with Retrieval-Augmented Generation (RAG) for dynamic information retrieval and response generation.
- C- An LLM dedicated to generating text responses without external data integration.
- D- A pre-trained LLM model from Cohere or OpenAI.

Answer:

B

Explanation:

Comprehensive and Detailed In-Depth Explanation=

For a chatbot needing to answer policy queries (requiring up-to-date, specific data) and retain chat history (context awareness), an LLM with RAG is ideal. RAG integrates external data (e.g., policy documents) via retrieval and supports memory for session-long context, making Option B correct. Option A (keyword search) lacks reasoning and context retention. Option C (standalone LLM) can't dynamically fetch policy data. Option D (pre-trained LLM) is too vague and lacks RAG's capabilities. RAG meets both requirements effectively.

: OCI 2025 Generative AI documentation likely highlights RAG for dynamic, context-aware applications.

Question 3

Question Type: MultipleChoice

Which is a characteristic of T-Few fine-tuning for Large Language Models (LLMs)?

Options:

- A- It updates all the weights of the model uniformly.
- B- It does not update any weights but restructures the model architecture.
- C- It selectively updates only a fraction of the model's weights.
- D- It increases the training time as compared to Vanilla fine-tuning.

Answer:

C

Explanation:

Comprehensive and Detailed In-Depth Explanation=

T-Few fine-tuning, a Parameter-Efficient Fine-Tuning (PEFT) method, updates only a small fraction of an LLM's weights, reducing computational cost and overfitting risk compared to Vanilla fine-tuning (all weights). This makes Option C correct. Option A describes Vanilla fine-tuning. Option B is false---T-Few updates weights, not architecture. Option D is incorrect---T-Few typically reduces training time. T-Few optimizes efficiency.

: OCI 2025 Generative AI documentation likely highlights T-Few under fine-tuning options.

Question 4

Question Type: MultipleChoice

How does a presence penalty function in language model generation when using OCI Generative AI service?

Options:

- A- It penalizes all tokens equally, regardless of how often they have appeared.
- B- It only penalizes tokens that have never appeared in the text before.
- C- It applies a penalty only if the token has appeared more than twice.
- D- It penalizes a token each time it appears after the first occurrence.

Answer:

D

Explanation:

Comprehensive and Detailed In-Depth Explanation=

A presence penalty in LLMs (including OCI's service) reduces the probability of tokens that have already appeared in the output, applying the penalty each time they reoccur after their first use. This discourages repetition, making Option D correct. Option A is false, as penalties depend on prior appearance, not uniform application. Option B is the opposite---penalizing unused tokens isn't the goal. Option C is incorrect, as the penalty isn't threshold-based (e.g., more than twice) but applied per reoccurrence. This enhances output diversity.

: OCI 2025 Generative AI documentation likely details presence penalty under generation parameters.

Question 5

Question Type: MultipleChoice

Which is a key characteristic of Large Language Models (LLMs) without Retrieval Augmented Generation (RAG)?

Options:

- A- They always use an external database for generating responses.
- B- They rely on internal knowledge learned during pretraining on a large text corpus.
- C- They cannot generate responses without fine-tuning.
- D- They use vector databases exclusively to produce answers.

Answer:

B

Explanation:

Comprehensive and Detailed In-Depth Explanation=

LLMs without Retrieval Augmented Generation (RAG) depend solely on the knowledge encoded in their parameters during pretraining on a large, general text corpus. They generate responses based on this internal knowledge without accessing external data at inference time, making Option B correct. Option A is false, as external databases are a feature of RAG, not standalone LLMs. Option C is incorrect, as LLMs can generate responses without fine-tuning via prompting or in-context learning. Option D is wrong, as vector databases are used in RAG or similar systems, not in basic LLMs. This reliance on pretraining distinguishes non-RAG LLMs from those augmented with real-time retrieval.

: OCI 2025 Generative AI documentation likely contrasts RAG and non-RAG LLMs under model architecture or response generation sections.

Question 6

Question Type: MultipleChoice

What does the Ranker do in a text generation system?

Options:

- A- It generates the final text based on the user's query.
- B- It sources information from databases to use in text generation.
- C- It evaluates and prioritizes the information retrieved by the Retriever.
- D- It interacts with the user to understand the query better.

Answer:

C

Explanation:

Comprehensive and Detailed In-Depth Explanation=

In systems like RAG, the Ranker evaluates and sorts the information retrieved by the Retriever (e.g., documents or snippets) based on relevance to the query, ensuring the most pertinent data is passed to the Generator. This makes Option C correct. Option A is the Generator's role. Option B describes the Retriever. Option D is unrelated, as the Ranker doesn't interact with users but processes retrieved data. The Ranker enhances output quality by prioritizing relevant content.

: OCI 2025 Generative AI documentation likely details the Ranker under RAG pipeline components.

Thank You for trying 1Z0-1127-25 PDF Demo

To try our 1Z0-1127-25 practice exam software
visit link below

<https://prepbolt.com/1Z0-1127-25.html>

Start Your 1Z0-1127-25 Preparation

Use Coupon "SAVE50" for extra 50% discount on the purchase of
Practice Test Software. Test your 1Z0-1127-25 preparation with actual
exam questions.