Free Questions for 1Z0-1127-25

Shared by White on 13-05-2025

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

What is the primary function of the "temperature" parameter in the OCI Generative AI Generation models?

Options:

A- Controls the randomness of the model's output, affecting its creativity
B- Specifies a string that tells the model to stop generating more content
C- Assigns a penalty to tokens that have already appeared in the preceding text
D- Determines the maximum number of tokens the model can generate per response

Answer:

A

Explanation:

Comprehensive and Detailed In-Depth Explanation=

The ''temperature'' parameter adjusts the randomness of an LLM's output by scaling the softmax distribution---low values (e.g., 0.7) make it more deterministic, high values (e.g., 1.5) increase creativity---Option A is correct. Option B (stop string) is the stop sequence. Option C (penalty) relates to presence/frequency penalties. Option D (max tokens) is a separate parameter. Temperature shapes output style.

: OCI 2025 Generative AI documentation likely defines temperature under generation parameters.

Question 2

Question Type: MultipleChoice

Which is NOT a built-in memory type in LangChain?

Options:

A- ConversationImageMemory

B- ConversationBufferMemory
C- ConversationSummaryMemory
D- ConversationTokenBufferMemory

Answer:

A

Explanation:

Comprehensive and Detailed In-Depth Explanation=

LangChain includes built-in memory types like ConversationBufferMemory (stores full history), ConversationSummaryMemory (summarizes history), and ConversationTokenBufferMemory (limits by token count)---Options B, C, and D are valid. ConversationImageMemory (A) isn't a standard type---image handling typically requires custom or multimodal extensions, not a built-in memory class---making A correct as NOT included.

: OCI 2025 Generative AI documentation likely lists memory types under LangChain memory management.

Question 3

Question Type: MultipleChoice

Given the following code:

chain = prompt | llm

Which statement is true about LangChain Expression Language (LCEL)?

Options:

A- LCEL is a programming language used to write documentation for LangChain.
B- LCEL is a legacy method for creating chains in LangChain.
C- LCEL is a declarative and preferred way to compose chains together.
D- LCEL is an older Python library for building Large Language Models.

Answer:

C

Explanation:

Comprehensive and Detailed In-Depth Explanation=

LangChain Expression Language (LCEL) is a declarative syntax (e.g., using | to pipe components) for composing chains in LangChain, combining prompts, LLMs, and other elements efficiently---Option C is correct. Option A is false---LCEL isn't for documentation. Option B is incorrect---it's current, not legacy; traditional Python classes are older. Option D is wrong---LCEL is part of LangChain, not a standalone LLM library. LCEL simplifies chain design.

: OCI 2025 Generative AI documentation likely highlights LCEL under LangChain chaincomposition.

Question 4

Question Type: MultipleChoice

Which statement is true about the "Top p" parameter of the OCI Generative AI Generation models?

Options:

A- 'Top p' selects tokens from the 'Top k' tokens sorted by probability.
B- 'Top p' assigns penalties to frequently occurring tokens.
C- 'Top p' limits token selection based on the sum of their probabilities.
D- 'Top p' determines the maximum number of tokens per response.

Answer:

C

Explanation:

Comprehensive and Detailed In-Depth Explanation=

''Top p'' (nucleus sampling) selects tokens whose cumulative probability exceeds a threshold (p), limiting the pool to the smallest set meeting this sum, enhancing diversity---Option C is correct. Option A confuses it with ''Top k.'' Option B (penalties) is unrelated. Option D (max tokens) is a different parameter. Top p balances randomness and coherence.

: OCI 2025 Generative AI documentation likely explains ''Top p'' under sampling methods.

Here is the next batch of 10 questions (81--90) from your list, formatted as requested with detailed explanations. The answers are based on widely accepted principles in generative AI and Large Language Models (LLMs), aligned with what is likely reflected in the Oracle Cloud Infrastructure (OCI) 2025 Generative AI documentation. Typographical errors have been corrected for clarity.

Question 5

Question Type: MultipleChoice

How are fine-tuned customer models stored to enable strong data privacy and security in the OCI Generative AI service?

Options:

A- Shared among multiple customers for efficiency
B- Stored in Object Storage encrypted by default
C- Stored in an unencrypted form in Object Storage
D- Stored in Key Management service

Answer:

B

Explanation:

Comprehensive and Detailed In-Depth Explanation=

In OCI, fine-tuned models are stored in Object Storage, encrypted by default, ensuring privacy and security per cloud best practices---Option B is correct. Option A (shared) violates privacy. Option C (unencrypted) contradicts security standards. Option D (Key Management) stores keys, not models. Encryption protects customer data.

: OCI 2025 Generative AI documentation likely details storage security under fine-tuning workflows.

Question 6

Question Type: MultipleChoice

Which is NOT a category of pretrained foundational models available in the OCI Generative AI service?

Options:

A- Summarization models
B- Generation models
C- Translation models
D- Embedding models

Answer:

C

Explanation:

Comprehensive and Detailed In-Depth Explanation=

OCI Generative AI typically offers pretrained models for summarization (A), generation (B), and embeddings (D), aligning with common generative tasks. Translation models (C) are less emphasized in generative AI services, often handled by specialized NLP platforms, making C the NOT category. While possible, translation isn't a core OCI generative focus based on standard offerings.

: OCI 2025 Generative AI documentation likely lists model categories under pretrained options.

Question 7

Question Type: MultipleChoice

What is the purpose of the "stop sequence" parameter in the OCI Generative AI Generation models?

Options:

A- It specifies a string that tells the model to stop generating more content.
B- It assigns a penalty to frequently occurring tokens to reduce repetitive text.
C- It determines the maximum number of tokens the model can generate per response.
D- It controls the randomness of the model's output, affecting its creativity.

Answer:

A

Explanation:

Comprehensive and Detailed In-Depth Explanation=

The ''stop sequence'' parameter defines a string (e.g., ''.'' or ''\n'') that, when generated, halts text generation, allowing control over output length or structure---Option A is correct. Option B (penalty) describes frequency/presence penalties. Option C (max tokens) is a separate parameter. Option D (randomness) relates to temperature. Stop sequences ensure precise termination.

: OCI 2025 Generative AI documentation likely details stop sequences under generation parameters.

Question 8

Question Type: MultipleChoice

What does a higher number assigned to a token signify in the "Show Likelihoods" feature of the language model token generation?

Options:

A- The token is less likely to follow the current token.
B- The token is more likely to follow the current token.
C- The token is unrelated to the current token and will not be used.
D- The token will be the only one considered in the next generation step.

Answer:

B

Explanation:

Comprehensive and Detailed In-Depth Explanation=

In ''Show Likelihoods,'' a higher number (probability score) indicates a token's greater likelihood of following the current token, reflecting the model's prediction confidence---Option B is correct.

Option A (less likely) is the opposite. Option C (unrelated) misinterprets---likelihood ties tokens contextually. Option D (only one) assumes greedy decoding, not the feature's purpose. This helps users understand model preferences.

: OCI 2025 Generative AI documentation likely explains ''Show Likelihoods'' under token generation insights.

Question 9

Question Type: MultipleChoice

Given the following code:

PromptTemplate(input_variables=["human_input", "city"], template=template)

Which statement is true about PromptTemplate in relation to input_variables?

Options:

A- PromptTemplate requires a minimum of two variables to function properly.
B- PromptTemplate can support only a single variable at a time.
C- PromptTemplate supports any number of variables, including the possibility of having none.
D- PromptTemplate is unable to use any variables.

Answer:

C

Explanation:

Comprehensive and Detailed In-Depth Explanation=

In LangChain, PromptTemplate supports any number of input_variables (zero, one, or more), allowing flexible prompt design---Option C is correct. The example shows two, but it's not a requirement. Option A (minimum two) is false---no such limit exists. Option B (single variable) is too restrictive. Option D (no variables) contradicts its purpose---variables are optional but supported. This adaptability aids prompt engineering.

: OCI 2025 Generative AI documentation likely covers PromptTemplate under LangChain prompt design.