



Vendor: Databricks

Exam Code: Generative-AI-Engineer-Associate

Exam Name: Databricks Certified Generative AI Engineer Associate

Version: DEMO

QUESTION 1

A Generative AI Engineer has been reviewing issues with their company's LLM based question-answering assistant and has determined that a technique called prompt chaining could help alleviate some performance concerns. However, to suggest this to their team, they have to clearly explain how it works and how it can benefit their question-answering assistant.

Which explanation do they communicate to the team?

- A. It allows you to break down complex tasks into multiple independent subtasks. This enables the assistant to generate more comprehensive and accurate responses.
- B. It allows you to reduce the latency of your applications. By having multiple chains participating in the response as a chain, you increase the rate at which the response is generated.
- C. It allows you to decrease the effort involved in crafting a prompt. Chains make it possible to reuse prompt text across multiple different use cases.
- D. It reduces the average cost of a typical request. Chains make more efficient use of the tokens produced to generate higher quality responses with fewer tokens.

Answer: A

Explanation:

Prompt chaining breaks down complex tasks into multiple smaller, sequential subtasks, allowing the assistant to handle each step methodically. This approach leads to more comprehensive and accurate responses by tackling each part of a problem in a structured way.

QUESTION 2

An AI developer team wants to fine tune an open-weight model to have exceptional performance on a code generation use case. They are trying to choose the best model to start with. They want to minimize model hosting costs, and are using Huggingface model cards and spaces to explore models.

Which TWO model attributes and metrics should the team focus on to make their selection? (Choose two.)

- A. Big Code Models Leaderboard
- B. Number of model parameters
- C. MTEB Leaderboard
- D. Chatbot Arena Leaderboard
- E. Number of model downloads last month

Answer: AB

Explanation:

The Big Code Models Leaderboard provides performance metrics specifically for code generation tasks, making it relevant for selecting a model optimized for this use case.

The number of model parameters directly impacts hosting costs and efficiency; smaller models reduce costs while balancing performance.

QUESTION 3

A small and cost-conscious startup in the cancer research field wants to build a RAG application using Foundation Model APIs.

Which strategy would allow the startup to build a good-quality RAG application while being cost-conscious and able to cater to customer needs?

- A. Limit the number of relevant documents available for the RAG application to retrieve from
- B. Pick a smaller LLM that is domain-specific
- C. Limit the number of queries a customer can send per day
- D. Use the largest LLM possible because that gives the best performance for any general queries

Answer: B

QUESTION 4

A Generative AI Engineer is deciding between using LSH (Locality Sensitive Hashing) and HNSW (Hierarchical Navigable Small World) for indexing their vector database. Their top priority is semantic accuracy.

Which approach should the Generative AI Engineer use to evaluate these two techniques?

- A. Compare the cosine similarities of the embeddings of returned results against those of a representative sample of test inputs
- B. Compare the Bilingual Evaluation Understudy (BLEU) scores of returned results for a representative sample of test inputs
- C. Compare the Recall-Oriented-Understudy for Gisting Evaluation (ROUGE) scores of returned results for a representative sample of test inputs
- D. Compare the Levenshtein distances of returned results against a representative sample of test inputs

Answer: A

Explanation:

Cosine similarity is a common metric used for evaluating the semantic accuracy of vector-based retrieval models, such as those used in LSH and HNSW indexing. By comparing the cosine similarities between the embeddings of returned results and the embeddings of a representative sample of test inputs, the Generative AI Engineer can assess how semantically accurate the results are in terms of their proximity to the correct or most relevant items.

QUESTION 5

A Generative AI Engineer has written scalable PySpark code to ingest unstructured PDF documents and chunk them in preparation for storing in a Databricks Vector Search index. Currently, the two columns of their dataframe include the original filename as a string and an array of text chunks from that document.

What set of steps should the Generative AI Engineer perform to store the chunks in a ready-to-ingest manner for Databricks Vector Search?

- A. Use PySpark's autoloader to apply a UDF across all chunks, formatting them in a JSON structure for Vector Search ingestion.
- B. Flatten the dataframe to one chunk per row, create a unique identifier for each row, and enable change feed on the output Delta table.
- C. Utilize the original filename as the unique identifier and save the dataframe as is.
- D. Create a unique identifier for each document, flatten the dataframe to one chunk per row and save to an output Delta table.

Answer: D

Explanation:

To prepare the data for Databricks Vector Search ingestion, the correct approach is to:

- Create a unique identifier for each document (e.g., a combination of the filename and an additional unique field to distinguish chunks from the same document).
- Flatten the dataframe so that each chunk has its own row, making it easier to index individual text chunks for vector search.
- Save the resulting dataframe into a Delta table to ensure the data is structured and ready for ingestion into the vector search index.

QUESTION 6

A Generative AI Engineer is developing a RAG system for their company to perform internal document Q&A for structured HR policies, but the answers returned are frequently incomplete and unstructured. It seems that the retriever is not returning all relevant context. The Generative AI Engineer has experimented with different embedding and response generating LLMs but that did not improve results.

Which TWO options could be used to improve the response quality? (Choose two.)

- A. Add the section header as a prefix to chunks
- B. Split the document by sentence
- C. Use a larger embedding model
- D. Increase the document chunk size
- E. Fine tune the response generation model

Answer: AE

Explanation:

Add the section header as a prefix to chunks: Adding the section header to the text chunks will provide more context to the retriever, allowing it to better understand the structure of the document. This context can help the model return more relevant and complete responses.

Fine-tune the response generation model: Fine-tuning the response generation model on the specific domain (in this case, HR policies) can improve the model's ability to provide structured and complete answers, tailored to the domain-specific language and requirements.

QUESTION 7

A Generative AI Engineer is developing a RAG application and would like to experiment with different embedding models to improve the application performance.

Which strategy for picking an embedding model should they choose?

- A. Pick an embedding model with multilingual support to support potential multilingual user questions
- B. Pick the most recent and most performant open LLM released at the time
- C. Pick an embedding model trained on related domain knowledge
- D. Pick the embedding model ranked highest on the Massive Text Embedding Benchmark (MTEB) leaderboard hosted by HuggingFace

Answer: C

Explanation:

For a RAG (Retrieve and Generate) application, the most effective embedding model would be one that is trained on domain-specific knowledge. This ensures that the model is familiar with the context and terminology related to the application, leading to better performance in retrieving relevant information and generating accurate responses.

QUESTION 8

Generative AI Engineer is helping a cinema extend its website's chat bot to be able to respond to questions about specific showtimes for movies currently playing at their local theater. They already have the location of the user provided by location services to their agent, and a Delta table which is continually updated with the latest showtime information by location. They want to implement this new capability in their RAG application.

Which option will do this with the least effort and in the most performant way?

- A. Create a Feature Serving Endpoint from a FeatureSpec that references an online store synced from the Delta table. Query the Feature Serving Endpoint as part of the agent logic / tool implementation.
- B. Query the Delta table directly via a SQL query constructed from the user's input using a text-to-SQL LLM in the agent logic / tool implementation.
- C. Set up a task in Databricks Workflows to write the information in the Delta table periodically to an external database such as MySQL and query the information from there as part of the agent logic / tool implementation.
- D. Write the Delta table contents to a text column, then embed those texts using an embedding model and store these in the vector index. Look up the information based on the embedding as part of the agent logic / tool implementation.

Answer: A

Explanation:

Chosen option is the most efficient and performant option for implementing the requested feature with the least amount of effort. Creating a **Feature Serving Endpoint** linked to the Delta table allows the chatbot to access up-to-date showtime information dynamically and with low latency, without needing additional transformations or external databases. The Feature Serving Endpoint will allow for efficient querying based on the user's location, making it easy to integrate with the agent's logic. This approach is specifically designed for serving feature data in production environments with minimal overhead.

QUESTION 9

Generative AI Engineer needs to build an LLM application that can understand medical documents, including recently published ones. They want to select an open model available on HuggingFace's model hub.

Which step is most appropriate for selecting an LLM?

- A. Pick any model in the Mistral family, as Mistral models are good with all types of use cases
- B. Select a model based on the highest number of downloads, as this indicates popularity, reliability, and general suitability
- C. Select a model that is most recently uploaded, as this indicates the model is the newest and highly likely to be the most performant
- D. Check for the model and training data description to identify if the model is trained on any medical data.

Answer: D

Explanation:

When selecting an LLM for understanding medical documents, it's crucial to ensure that the model is trained or fine-tuned on domain-specific data, such as medical text. The model's performance on medical tasks will be highly dependent on whether it has been trained with

relevant data (e.g., clinical notes, medical literature, etc.). Therefore, reviewing the model's training data description is the most appropriate step to determine if it is suitable for understanding medical documents.

QUESTION 10

Generative AI Engineer is building a RAG application that answers questions about technology-related news articles. The source documents may contain a significant amount of irrelevant content, such as advertisements, sports news, or entertainment news.

Which approach is **NOT** advisable for building a RAG application focused on answering technology-only questions?

- A. Include in the system prompt that the application is not supposed to answer any questions unrelated to technology.
- B. Filter out irrelevant news articles in the retrieval process.
- C. Keep all news articles because the RAG application needs to understand non-technological content to avoid answering questions about them.
- D. Filter out irrelevant news articles in the upstream document database.

Answer: C

Explanation:

Chosen option is not advisable because keeping all news articles, including irrelevant ones, would increase the volume of unnecessary data in the retrieval process. This could reduce the efficiency and accuracy of the RAG application by introducing irrelevant content into the context provided for answering technology-related questions. It could also lead to increased computational costs, as the system would need to process a larger dataset with more noise.

QUESTION 11

A Generative AI Engineer is building a RAG application that will rely on context retrieved from source documents that are currently in HTML format. They want to develop a solution using the least amount of lines of code.

Which Python package should be used to extract the text from the source documents?

- A. pytesseract
- B. numpy
- C. pypdf2
- D. BeautifulSoup

Answer: D

Explanation:

BeautifulSoup is a Python package specifically designed for parsing HTML and XML documents. It provides simple methods to extract and manipulate text from HTML files, making it the most suitable choice for extracting text from HTML source documents with minimal lines of code.

Thank You for Trying Our Product

Lead2pass Certification Exam Features:

- ★ More than **99,900** Satisfied Customers Worldwide.
- ★ Average **99.9%** Success Rate.
- ★ **Free Update** to match latest and real exam scenarios.
- ★ **Instant Download** Access! No Setup required.
- ★ Questions & Answers are downloadable in **PDF** format and **VCE** test engine format.
- ★ Multi-Platform capabilities - **Windows, Laptop, Mac, Android, iPhone, iPod, iPad**.
- ★ **100%** Guaranteed Success or **100%** Money Back Guarantee.
- ★ **Fast**, helpful support **24x7**.



View list of all certification exams: <http://www.lead2pass.com/all-products.html>



Microsoft



ORACLE



JUNIPER
NETWORKS



EMC²
where information lives

10% Discount Coupon Code: ASTR14

