Databricks.Databricks-Generative-AI-Engineer-Associate.v2025-02-03.q15

Exam Code:	Databricks-Generative-AI-Engineer-Associate
Exam Name:	Databricks Certified Generative AI Engineer Associate
Certification Provider:	Databricks
Free Question Number:	15
Version:	v2025-02-03
# of views:	321
# of Questions views:	150
https://www.freeqas.com/qa/Databricks/Databricks-Generative-AI-Engineer-Associate/Databricks.Databricks-Generative-AI-Engineer-Associate.v2025-02-03.q15.html	

NEW QUESTION: 1
A Generative AI Engineer is creating an agent-based LLM system for their favorite monster truck team. The system can answer text based questions about the monster truck team, lookup event dates via an API call, or query tables on the team's latest standings.
How could the Generative AI Engineer best design these capabilities into their system?
A. Ingest PDF documents about the monster truck team into a vector store and query it in a RAG architecture.
B. Write a system prompt for the agent listing available tools and bundle it into an agent system that runs a number of calls to solve a query.
C. Instruct the LLM to respond with "RAG", "API", or "TABLE" depending on the query, then use text parsing and conditional statements to resolve the query.
D. Build a system prompt with all possible event dates and table information in the system prompt. Use a RAG architecture to lookup generic text questions and otherwise leverage the information in the system prompt.
Answer: B (LEAVE A REPLY)
In this scenario, the Generative AI Engineer needs to design a system that can handle different types of queries about the monster truck team. The queries may involve text-based information, API lookups for event dates, or table queries for standings. The best solution is to implement atool-based agent system.
Here's how option B works, and why it's the most appropriate answer:
* System Design Using Agent-Based Model:In modern agent-based LLM systems, you can design a system where the LLM (Large Language Model) acts as a central orchestrator. The model can "decide" which tools to use based on the query. These tools can include

API calls, table lookups, or natural language searches. The system should contain asystem promptthat informs the LLM about the available tools.
* System Prompt Listing Tools:By creating a well-craftedsystem prompt, the LLM knows which tools are at its disposal. For instance, one tool may query an external API for event dates, another might look up standings in a database, and a third may involve searching a vector database for general text-based information. Theagentwill be responsible for calling the appropriate tool depending on the query.
* Agent Orchestration of Calls:The agent system is designed to execute a series of steps based on the incoming query. If a user asks for the next event date, the system will recognize this as a task that requires an API call. If the user asks about standings, the agent might query the appropriate table in the database. For text-based questions, it may call a search function over ingested data. The agent orchestrates this entire process, ensuring the LLM makes calls to the right resources dynamically.
* Generative AI Tools and Context:This is a standard architecture for integrating multiple functionalities into a system where each query requires different actions. The core design in option B is efficient because it keeps the system modular and dynamic by leveraging tools rather than overloading the LLM with static information in a system prompt (like option D).
* Why Other Options Are Less Suitable:
* A (RAG Architecture): While relevant, simply ingesting PDFs into a vector store only helps with text-based retrieval. It wouldn't help with API lookups or table queries.
* C (Conditional Logic with RAG/API/TABLE): Although this approach works, it relies heavily on manual text parsing and might introduce complexity when scaling the system.
* D (System Prompt with Event Dates and Standings): Hardcoding dates and table information into a system prompt isn't scalable. As the standings or events change, the system would need constant updating, making it inefficient.
By bundling multiple tools into a single agent-based system (as in option B), the Generative AI Engineer can best handle the diverse requirements of this system.

NEW QUESTION: 2
A Generative AI Engineer has been asked to design an LLM-based application that accomplishes the following business objective: answer employee HR questions using HR PDF documentation.
Which set of high level tasks should the Generative AI Engineer's system perform?
A. Calculate averaged embeddings for each HR document, compare embeddings to user query to find the best document. Pass the best document with the user query into an LLM with a large context window to generate a response to the employee.
B. Use an LLM to summarize HR documentation. Provide summaries of documentation and user query into an LLM with a large context window to generate a response to the user.

C. Create an interaction matrix of historical employee questions and HR documentation. Use ALS to factorize the matrix and create embeddings. Calculate the embeddings of new queries and use them to find the best HR documentation. Use an LLM to generate a response to the employee question based upon the documentation retrieved.
D. Split HR documentation into chunks and embed into a vector store. Use the employee question to retrieve best matched chunks of documentation, and use the LLM to generate a response to the employee based upon the documentation retrieved.
Answer: (SHOW ANSWER)
To design an LLM-based application that can answer employee HR questions using HR PDF documentation, the most effective approach is option D. Here's why:
* Chunking and Vector Store Embedding:HR documentation tends to be lengthy, so splitting it into smaller, manageable chunks helps optimize retrieval. These chunks are then embedded into avector store(a database that stores vector representations of text). Each chunk of text is transformed into an embeddingusing a transformer-based model, which allows for efficient similarity-based retrieval.
* Using Vector Search for Retrieval:When an employee asks a question, the system converts their query into an embedding as well. This embedding is then compared with the embeddings of the document chunks in the vector store. The most semantically similar chunks are retrieved, which ensures that the answer is based on the most relevant parts of the documentation.
* LLM to Generate a Response:Once the relevant chunks are retrieved, these chunks are passed into the LLM, which uses them as context to generate a coherent and accurate response to the employee's question.
* Why Other Options Are Less Suitable:
* A (Calculate Averaged Embeddings): Averaging embeddings might dilute important information. It doesn't provide enough granularity to focus on specific sections of documents.
* B (Summarize HR Documentation): Summarization loses the detail necessary for HR-related queries, which are often specific. It would likely miss the mark for more detailed inquiries.
* C (Interaction Matrix and ALS): This approach is better suited for recommendation systems and not for HR queries, as it's focused on collaborative filtering rather than text-based retrieval.
Thus, option D is the most effective solution for providing precise and contextual answers based on HR documentation.

NEW QUESTION: 3
A Generative AI Engineer has been asked to build an LLM-based question-answering application. The application should take into account new documents that are frequently published. The engineer wants to build this application with the least cost and least development effort and have it operate at the lowest cost possible.

Which combination of chaining components and configuration meets these requirements?
A. For the application a prompt, a retriever, and an LLM are required. The retriever output is inserted into the prompt which is given to the LLM to generate answers.
B. The LLM needs to be frequently with the new documents in order to provide most up-to-date answers.
C. For the question-answering application, prompt engineering and an LLM are required to generate answers.
D. For the application a prompt, an agent and a fine-tuned LLM are required. The agent is used by the LLM to retrieve relevant content that is inserted into the prompt which is given to the LLM to generate answers.
Answer: A (LEAVE A REPLY)
Problem Context: The task is to build an LLM-based question-answering application that integrates new documents frequently with minimal costs and development efforts.
Explanation of Options:
* Option A: Utilizes a prompt and a retriever, with the retriever output being fed into the LLM. This setup is efficient because it dynamically updates the data pool via the retriever, allowing the LLM to provide up-to-date answers based on the latest documents without needing tofrequently retrain the model. This method offers a balance of cost-effectiveness and functionality.
* Option B: Requires frequent retraining of the LLM, which is costly and labor-intensive.
* Option C: Only involves prompt engineering and an LLM, which may not adequately handle the requirement for incorporating new documents unless it's part of an ongoing retraining or updating mechanism, which would increase costs.
* Option D: Involves an agent and a fine-tuned LLM, which could be overkill and lead to higher development and operational costs.
Option Ais the most suitable as it provides a cost-effective, minimal development approach while ensuring the application remains up-to-date with new information.

NEW QUESTION: 4
A Generative AI Engineer has developed an LLM application to answer questions about internal company policies. The Generative AI Engineer must ensure that the application doesn't hallucinate or leak confidential data.
Which approach should NOT be used to mitigate hallucination or confidential data leakage?
A. Add guardrails to filter outputs from the LLM before it is shown to the user
B. Fine-tune the model on your data, hoping it will learn what is appropriate and not
C. Limit the data available based on the user's access level
D. Use a strong system prompt to ensure the model aligns with your needs.
Answer: B (LEAVE A REPLY)

When addressing concerns of hallucination and data leakage in an LLM application for internal company policies, fine-tuning the model on internal data with the hope it learns data boundaries can be problematic:
* Risk of Data Leakage: Fine-tuning on sensitive or confidential data does not guarantee that the model will not inadvertently include or reference this data in its outputs. There's a risk of overfitting to the specific data details, which might lead to unintended leakage.
* Hallucination: Fine-tuning does not necessarily mitigate the model's tendency to hallucinate; in fact, it might exacerbate it if the training data is not comprehensive or representative of all potential queries.
Better Approaches:
* A,C, andDinvolve setting up operational safeguards and constraints that directly address data leakage and ensure responses are aligned with specific user needs and security levels.
Fine-tuning lacks the targeted control needed for such sensitive applications and can introduce new risks, making it an unsuitable approach in this context.

NEW QUESTION: 5
A Generative AI Engineer is developing a chatbot designed to assist users with insurance-related queries. The chatbot is built on a large language model (LLM) and is conversational. However, to maintain the chatbot's focus and to comply with company policy, it must not provide responses to questions about politics. Instead, when presented with political inquiries, the chatbot should respond with a standard message:
"Sorry, I cannot answer that. I am a chatbot that can only answer questions around insurance." Which framework type should be implemented to solve this?
A. Safety Guardrail
B. Security Guardrail
C. Contextual Guardrail
D. Compliance Guardrail
Answer: A (LEAVE A REPLY)
In this scenario, the chatbot must avoid answering political questions and instead provide a standard message for such inquiries. Implementing aSafety Guardrailis the appropriate solution for this:
* What is a Safety Guardrail?Safety guardrails are mechanisms implemented in Generative AI systems to ensure the model behaves within specific bounds. In this case, it ensures the chatbot does not answer politically sensitive or irrelevant questions, which aligns with the business rules.
* Preventing Responses to Political Questions:The Safety Guardrail is programmed to detect specific types of inquiries (like political questions) and prevent the model from generating responses outside its intended domain. When such queries are detected, the guardrail intervenes and provides a pre-defined response: "Sorry, I cannot answer that. I am a chatbot that can only answer questions around insurance."

* How It Works in Practice:The LLM system can include aclassification layeror trigger rules based on specific keywords related to politics. When such terms are detected, the Safety Guardrail blocks the normal generation flow and responds with the fixed message.
* Why Other Options Are Less Suitable:
* B (Security Guardrail): This is more focused on protecting the system from security vulnerabilities or data breaches, not controlling the conversational focus.
* C (Contextual Guardrail): While context guardrails can limit responses based on context, safety guardrails are specifically about ensuring the chatbot stays within a safe conversational scope.
* D (Compliance Guardrail): Compliance guardrails are often related to legal and regulatory adherence, which is not directly relevant here.
Therefore, aSafety Guardrailis the right framework to ensure the chatbot only answers insurance-related queries and avoids political discussions.

NEW QUESTION: 6
A Generative AI Engineer is tasked with improving the RAG quality by addressing its inflammatory outputs.
Which action would be most effective in mitigating the problem of offensive text outputs?
A. Increase the frequency of upstream data updates
B. Inform the user of the expected RAG behavior
C. Restrict access to the data sources to a limited number of users
D. Curate upstream data properly that includes manual review before it is fed into the RAG system
Answer: D (LEAVE A REPLY)
Addressing offensive or inflammatory outputs in a Retrieval-Augmented Generation (RAG) system is critical for improving user experience and ensuring ethical AI deployment. Here's whyDis the most effective approach:
* Manual data curation: The root cause of offensive outputs often comes from the underlying data used to train the model or populate the retrieval system. By manually curating the upstream data and conducting thorough reviews before the data is fed into the RAG system, the engineer can filter out harmful, offensive, or inappropriate content.
* Improving data quality: Curating data ensures the system retrieves and generates responses from a high-quality, well-vetted dataset. This directly impacts the relevance and appropriateness of the outputs from the RAG system, preventing inflammatory content from being included in responses.
* Effectiveness: This strategy directly tackles the problem at its source (the data) rather than just mitigating the consequences (such as informing users or restricting access). It ensures that the system consistently provides non-offensive, relevant information.
Other options, such as increasing the frequency of data updates or informing users about behavior expectations, may not directly mitigate the generation of inflammatory outputs.

NEW QUESTION: 7
A Generative AI Engineer is designing a RAG application for answering user questions on technical regulations as they learn a new sport.
What are the steps needed to build this RAG application and deploy it?
A. Ingest documents from a source -> Index the documents and saves to Vector Search -> User submits queries against an LLM -> LLM retrieves relevant documents -> Evaluate model -> LLM generates a response -> Deploy it using Model Serving
B. Ingest documents from a source -> Index the documents and save to Vector Search -> User submits queries against an LLM -> LLM retrieves relevant documents -> LLM generates a response -> Evaluate model -> Deploy it using Model Serving
C. Ingest documents from a source -> Index the documents and save to Vector Search -> Evaluate model -
> Deploy it using Model Serving
D. User submits queries against an LLM -> Ingest documents from a source -> Index the documents and save to Vector Search -> LLM retrieves relevant documents -> LLM generates a response -> Evaluate model -> Deploy it using Model Serving
Answer: B (LEAVE A REPLY)
The Generative AI Engineer needs to follow a methodical pipeline to build and deploy a Retrieval- Augmented Generation (RAG) application. The steps outlined in optionBaccurately reflect this process:
* Ingest documents from a source: This is the first step, where the engineer collects documents (e.g., technical regulations) that will be used for retrieval when the application answers user questions.
* Index the documents and save to Vector Search: Once the documents are ingested, they need to be embedded using a technique like embeddings (e.g., with a pre-trained model like BERT) and stored in a vector database (such as Pinecone or FAISS). This enables fast retrieval based on user queries.
* User submits queries against an LLM: Users interact with the application by submitting their queries.
These queries will be passed to the LLM.
* LLM retrieves relevant documents: The LLM works with the vector store to retrieve the most relevant documents based on their vector representations.
* LLM generates a response: Using the retrieved documents, the LLM generates a response that is tailored to the user's question.
* Evaluate model: After generating responses, the system must be evaluated to ensure the retrieved documents are relevant and the generated response is accurate. Metrics such as accuracy, relevance, and user satisfaction can be used for evaluation.
* Deploy it using Model Serving: Once the RAG pipeline is ready and evaluated, it is deployed using a model-serving platform such as Databricks Model Serving. This enables real-time inference and response generation for users.

By following these steps, the Generative AI Engineer ensures that the RAG application is both efficient and effective for the task of answering technical regulation questions.

NEW QUESTION: 8
A Generative AI Engineer is creating an LLM-based application. The documents for its retriever have been chunked to a maximum of 512 tokens each. The Generative AI Engineer knows that cost and latency are more important than quality for this application. They have several context length levels to choose from.
Which will fulfill their need?
A. context length 514; smallest model is 0.44GB and embedding dimension 768
B. context length 2048: smallest model is 11GB and embedding dimension 2560
C. context length 32768: smallest model is 14GB and embedding dimension 4096
D. context length 512: smallest model is 0.13GB and embedding dimension 384
Answer: D (LEAVE A REPLY)
When prioritizing cost and latency over quality in a Large Language Model (LLM)-based application, it is crucial to select a configuration that minimizes both computational resources and latency while still providing reasonable performance. Here's whyDis the best choice:
* Context length: The context length of 512 tokens aligns with the chunk size used for the documents (maximum of 512 tokens per chunk). This is sufficient for capturing the needed information and generating responses without unnecessary overhead.
* Smallest model size: The model with a size of 0.13GB is significantly smaller than the other options.
This small footprint ensures faster inference times and lower memory usage, which directly reduces both latency and cost.
* Embedding dimension: While the embedding dimension of 384 is smaller than the other options, it is still adequate for tasks where cost and speed are more important than precision and depth of understanding.
This setup achieves the desired balance between cost-efficiency and reasonable performance in a latency- sensitive, cost-conscious application.

NEW QUESTION: 9
A Generative AI Engineer is designing a chatbot for a gaming company that aims to engage users on its platform while its users play online video games.
Which metric would help them increase user engagement and retention for their platform?
A. Randomness
B. Diversity of responses
C. Lack of relevance
D. Repetition of responses
Answer: B (LEAVE A REPLY)

In the context of designing a chatbot to engage users on a gaming platform,diversity of responses(option B) is a key metric to increase user engagement and retention. Here's why:
* Diverse and Engaging Interactions:A chatbot that provides varied and interesting responses will keep users engaged, especially in an interactive environment like a gaming platform. Gamers typically enjoy dynamic and evolving conversations, anddiversity of responseshelps prevent monotony, encouraging users to interact more frequently with the bot.
* Increasing Retention:By offering different types of responses to similar queries, the chatbot can create a sense of novelty and excitement, which enhances the user's experience and makes them more likely to return to the platform.
* Why Other Options Are Less Effective:
* A (Randomness): Random responses can be confusing or irrelevant, leading to frustration and reducing engagement.
* C (Lack of Relevance): If responses are not relevant to the user's queries, this will degrade the user experience and lead to disengagement.
* D (Repetition of Responses): Repetitive responses can quickly bore users, making the chatbot feel uninteresting and reducing the likelihood of continued interaction.
Thus,diversity of responses(option B) is the most effective way to keep users engaged and retain them on the platform.

NEW QUESTION: 10
A Generative AI Engineer just deployed an LLM application at a digital marketing company that assists with answering customer service inquiries.
Which metric should they monitor for their customer service LLM application in production?
A. Number of customer inquiries processed per unit of time
B. Energy usage per query
C. Final perplexity scores for the training of the model
D. HuggingFace Leaderboard values for the base LLM
Answer: A (LEAVE A REPLY)
When deploying an LLM application for customer service inquiries, the primary focus is on measuring the operational efficiency and quality of the responses. Here's whyAis the correct metric:
* Number of customer inquiries processed per unit of time: This metric tracks the throughput of the customer service system, reflecting how many customer inquiries the LLM application can handle in a given time period (e.g., per minute or hour). High throughput is crucial in customer service applications where quick response times are essential to user satisfaction and business efficiency.
* Real-time performance monitoring: Monitoring the number of queries processed is an important part of ensuring that the model is performing well under load, especially during peak traffic times. It also helps ensure the system scales properly to meet demand.

Why other options are not ideal:
* B. Energy usage per query: While energy efficiency is a consideration, it is not the primary concern for a customer-facing application where user experience (i.e., fast and accurate responses) is critical.
* C. Final perplexity scores for the training of the model: Perplexity is a metric for model training, but it doesn't reflect the real-time operational performance of an LLM in production.
* D. HuggingFace Leaderboard values for the base LLM: The HuggingFace Leaderboard is more relevant during model selection and benchmarking. However, it is not a direct measure of the model's performance in a specific customer service application in production.
Focusing on throughput (inquiries processed per unit time) ensures that the LLM application is meeting business needs for fast and efficient customer service responses.

NEW QUESTION: 11
A Generative AI Engineer has created a RAG application to look up answers to questions about a series of fantasy novels that are being asked on the author's web forum. The fantasy novel texts are chunked and embedded into a vector store with metadata (page number, chapter number, book title), retrieved with the user' s query, and provided to an LLM for response generation. The Generative AI Engineer used their intuition to pick the chunking strategy and associated configurations but now wants to more methodically choose the best values.
Which TWO strategies should the Generative AI Engineer take to optimize their chunking strategy and parameters? (Choose two.)
A. Change embedding models and compare performance.
B. Add a classifier for user queries that predicts which book will best contain the answer. Use this to filter retrieval.
C. Choose an appropriate evaluation metric (such as recall or NDCG) and experiment with changes in the chunking strategy, such as splitting chunks by paragraphs or chapters. Choose the strategy that gives the best performance metric.
D. Pass known questions and best answers to an LLM and instruct the LLM to provide the best token count. Use a summary statistic (mean, median, etc.) of the best token counts to choose chunk size.
E. Create an LLM-as-a-judge metric to evaluate how well previous questions are answered by the most appropriate chunk. Optimize the chunking parameters based upon the values of the metric.
Answer: (SHOW ANSWER)
To optimize a chunking strategy for a Retrieval-Augmented Generation (RAG) application, the Generative AI Engineer needs a structured approach to evaluating the chunking strategy, ensuring that the chosen configuration retrieves the most relevant information and leads to accurate and coherent LLM responses.

Here's whyCandEare the correct strategies:
Strategy C: Evaluation Metrics (Recall, NDCG)
* Define an evaluation metric: Common evaluation metrics such as recall, precision, or NDCG (Normalized Discounted Cumulative Gain) measure how well the retrieved chunks match the user's query and the expected response.
* Recallmeasures the proportion of relevant information retrieved.
* NDCGis often used when you want to account for both the relevance of retrieved chunks and the ranking or order in which they are retrieved.
* Experiment with chunking strategies: Adjusting chunking strategies based on text structure (e.g., splitting by paragraph, chapter, or a fixed number of tokens) allows the engineer to experiment with various ways of slicing the text. Some chunks may better align with the user's query than others.
* Evaluate performance: By using recall or NDCG, the engineer can methodically test various chunking strategies to identify which one yields the highest performance. This ensures that the chunking method provides the most relevant information when embedding and retrieving data from the vector store.
Strategy E: LLM-as-a-Judge Metric
* Use the LLM as an evaluator: After retrieving chunks, the LLM can be used to evaluate the quality of answers based on the chunks provided. This could be framed as a "judge" function, where the LLM compares how well a given chunk answers previous user queries.
* Optimize based on the LLM's judgment: By having the LLM assess previous answers and rate their relevance and accuracy, the engineer can collect feedback on how well different chunking configurations perform in real-world scenarios.
* This metric could be a qualitative judgment on how closely the retrieved information matches the user's intent.
* Tune chunking parameters: Based on the LLM's judgment, the engineer can adjust the chunk size or structure to better align with the LLM's responses, optimizing retrieval for future queries.
By combining these two approaches, the engineer ensures that the chunking strategy is systematically evaluated using both quantitative (recall/NDCG) and qualitative (LLM judgment) methods. This balanced optimization process results in improved retrieval relevance and, consequently, better response generation by the LLM.

NEW QUESTION: 12
A Generative Al Engineer is building a RAG application that answers questions about internal documents for the company SnoPen AI.
The source documents may contain a significant amount of irrelevant content, such as advertisements, sports news, or entertainment news, or content about other companies.
Which approach is advisable when building a RAG application to achieve this goal of filtering irrelevant information?

A. Keep all articles because the RAG application needs to understand non-company content to avoid answering questions about them.
B. Include in the system prompt that any information it sees will be about SnoPenAI, even if no data filtering is performed.
C. Include in the system prompt that the application is not supposed to answer any questions unrelated to SnoPen AI.
D. Consolidate all SnoPen AI related documents into a single chunk in the vector database.
Answer: C (LEAVE A REPLY)
In a Retrieval-Augmented Generation (RAG) application built to answer questions about internal documents, especially when the dataset contains irrelevant content, it's crucial to guide the system to focus on the right information. The best way to achieve this is byincluding a clear instruction in the system prompt(option C).
* System Prompt as Guidance:The system prompt is an effective way to instruct the LLM to limit its focus to SnoPen AI-related content. By clearly specifying that the model should avoid answering questions unrelated to SnoPen AI, you add an additional layer of control that helps the model stay on- topic, even if irrelevant content is present in the dataset.
* Why This Approach Works:The prompt acts as a guiding principle for the model, narrowing its focus to specific domains. This prevents the model from generating answers based on irrelevant content, such as advertisements or news unrelated to SnoPen AI.
* Why Other Options Are Less Suitable:
* A (Keep All Articles): Retaining all content, including irrelevant materials, without any filtering makes the system prone to generating answers based on unwanted data.
* B (Include in the System Prompt about SnoPen AI): This option doesn't address irrelevant content directly, and without filtering, the model might still retrieve and use irrelevant data.
* D (Consolidating Documents into a Single Chunk): Grouping documents into a single chunk makes the retrieval process less efficient and won't help filter out irrelevant content effectively.
Therefore, instructing the system in the prompt not to answer questions unrelated to SnoPen AI (option C) is the best approach to ensure the system filters out irrelevant information.

NEW QUESTION: 13
A Generative AI Engineer is building an LLM to generate article summaries in the form of a type of poem, such as a haiku, given the article content. However, the initial output from the LLM does not match the desired tone or style.
Which approach will NOT improve the LLM's response to achieve the desired response?
A. Provide the LLM with a prompt that explicitly instructs it to generate text in the desired tone and style
B. Use a neutralizer to normalize the tone and style of the underlying documents

C. Include few-shot examples in the prompt to the LLM
D. Fine-tune the LLM on a dataset of desired tone and style
Answer: B (LEAVE A REPLY)
The task at hand is to improve the LLM's ability to generate poem-like article summaries with the desired tone and style. Using aneutralizerto normalize the tone and style of the underlying documents (option B) will not help improve the LLM's ability to generate the desired poetic style. Here's why:
* Neutralizing Underlying Documents:A neutralizer aims to reduce or standardize the tone of input data. However, this contradicts the goal, which is to generate text with aspecific tone and style(like haikus). Neutralizing the source documents will strip away the richness of the content, making it harder for the LLM to generate creative, stylistic outputs like poems.
* Why Other Options Improve Results:
* A (Explicit Instructions in the Prompt): Directly instructing the LLM to generate text in a specific tone and style helps align the output with the desired format (e.g., haikus). This is a common and effective technique in prompt engineering.
* C (Few-shot Examples): Providing examples of the desired output format helps the LLM understand the expected tone and structure, making it easier to generate similar outputs.
* D (Fine-tuning the LLM): Fine-tuning the model on a dataset that contains examples of the desired tone and style is a powerful way to improve the model's ability to generate outputs that match the target format.
Therefore, using a neutralizer (option B) isnotan effective method for achieving the goal of generating stylized poetic summaries.

NEW QUESTION: 14
A Generative AI Engineer has created a RAG application which can help employees retrieve answers from an internal knowledge base, such as Confluence pages or Google Drive. The prototype application is now working with some positive feedback from internal company testers. Now the Generative AI Engineer wants to formally evaluate the system's performance and understand where to focus their efforts to further improve the system. How should the Generative AI Engineer evaluate the system?
A. Use cosine similarity score to comprehensively evaluate the quality of the final generated answers.
B. Curate a dataset that can test the retrieval and generation components of the system separately. Use MLflow's built in evaluation metrics to perform the evaluation on the retrieval and generation components.
C. Benchmark multiple LLMs with the same data and pick the best LLM for the job.
D. Use an LLM-as-a-judge to evaluate the quality of the final answers generated.
Answer: (SHOW ANSWER)
* Problem Context: After receiving positive feedback for the RAG application prototype, the next step is to formally evaluate the system to pinpoint areas for improvement.

* Explanation of Options:
* Option A: While cosine similarity scores are useful, they primarily measure similarity rather than the overall performance of an RAG system.
* Option B: This option provides a systematic approach to evaluation by testing both retrieval and generation components separately. This allows for targeted improvements and a clear understanding of each component's performance, using MLflow's metrics for a structured and standardized assessment.
* Option C: Benchmarking multiple LLMs does not focus on evaluating the existing system's components but rather on comparing different models.
* Option D: Using an LLM as a judge is subjective and less reliable for systematic performance evaluation.
OptionBis the most comprehensive and structured approach, facilitating precise evaluations and improvements on specific components of the RAG system.

NEW QUESTION: 15
When developing an LLM application, it's crucial to ensure that the data used for training the model complies with licensing requirements to avoid legal risks.
Which action is NOT appropriate to avoid legal risks?
A. Reach out to the data curators directly before you have started using the trained model to let them know.
B. Use any available data you personally created which is completely original and you can decide what license to use.
C. Only use data explicitly labeled with an open license and ensure the license terms are followed.
D. Reach out to the data curators directly after you have started using the trained model to let them know.
Answer: D (LEAVE A REPLY)
* Problem Context: When using data to train a model, it's essential to ensure compliance with licensing to avoid legal risks. Legal issues can arise from using data without permission, especially when it comes from third-party sources.
* Explanation of Options:
* Option A: Reaching out to data curatorsbeforeusing the data is an appropriate action. This allows you to ensure you have permission or understand the licensing terms before starting to use the data in your model.
* Option B: Usingoriginal datathat you personally created is always a safe option. Since you have full ownership over the data, there are no legal risks, as you control the licensing.
* Option C: Using data that is explicitly labeled with an open license and adhering to the license terms is a correct and recommended approach. This ensures compliance with legal requirements.
* Option D: Reaching out to the data curatorsafteryou have already started using the trained model isnot appropriate. If you've already used the data without understanding its

licensing terms, you may have already violated the terms of use, which could lead to legal complications. It's essential to clarify the licensing termsbeforeusing the data, not after. Thus,Option Dis not appropriate because it could expose you to legal risks by using the data without first obtaining the proper licensing permissions.