Free Questions for Databricks-Generative-AI-Engineer-Associate

Shared by Keller on 30-09-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: **MultipleChoice**

A Generative AI Engineer is tasked with developing an application that is based on an open source large language model (LLM). They need a foundation LLM with a large context window.

Which model fits this need?

Options:

A- DistilBERT

B- MPT-30B

C- Llama2-70B

D- DBRX

Answer:

C

Explanation:

Problem Context: The engineer needs an open-source LLM with a large context window to develop an application.

Explanation of Options:

Option A: DistilBERT: While an efficient and smaller version of BERT, DistilBERT does not provide a particularly large context window.

Option B: MPT-30B: This model, while large, is not specified as being particularly notable for its context window capabilities.

Option C: Llama2-70B: Known for its large model size and extensive capabilities, including a large context window. It is also available as an open-source model, making it ideal for applications requiring extensive contextual understanding.

Option D: DBRX: This is not a recognized standard model in the context of large language models with extensive context windows.

Thus, Option C (Llama2-70B) is the best fit as it meets the criteria of having a large context window and being available for open-source use, suitable for developing robust language understanding applications.

Question 2

Question Type: MultipleChoice

A Generative AI Engineer received the following business requirements for an external chatbot.

The chatbot needs to know what types of questions the user asks and routes to appropriate models to answer the questions. For example, the user might ask about upcoming event details. Another user might ask about purchasing tickets for a particular event.

What is an ideal workflow for such a chatbot?

Options:

A- The chatbot should only look at previous event information

B- There should be two different chatbots handling different types of user queries.

C- The chatbot should be implemented as a multi-step LLM workflow. First, identify the type of question asked, then route the question to the appropriate model. If it's an upcoming event question, send the query to a text-to-SQL model. If it's about ticket purchasing, the customer should be redirected to a payment platform.

D- The chatbot should only process payments

Answer:

C

Explanation:

Problem Context: The chatbot must handle various types of queries and intelligently route them to the appropriate responses or systems.

Explanation of Options:

Option A: Limiting the chatbot to only previous event information restricts its utility and does not meet the broader business requirements.

Option B: Having two separate chatbots could unnecessarily complicate user interaction and increase maintenance overhead.

Option C: Implementing a multi-step workflow where the chatbot first identifies the type of question and then routes it accordingly is the most efficient and scalable solution. This approach allows the chatbot to handle a variety of queries dynamically, improving user experience and operational efficiency.

Option D: Focusing solely on payments would not satisfy all the specified user interaction needs, such as inquiring about event details.

Option C offers a comprehensive workflow that maximizes the chatbot's utility and responsiveness to different user needs, aligning perfectly with the business requirements.

Question 3

Question Type: MultipleChoice

A Generative AI Engineer is building a RAG application that will rely on context retrieved from source documents that are currently in PDF format. These PDFs can contain both text and images. 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?

Options:

A- flask

B- beautifulsoup

C- unstructured

D- numpy

Answer:

C

Explanation:

Problem Context: The engineer needs to extract text from PDF documents, which may contain both text and images. The goal is to find a Python package that simplifies this task using the least amount of code.

Explanation of Options:

Option A: flask: Flask is a web framework for Python, not suitable for processing or extracting content from PDFs.

Option B: beautifulsoup: Beautiful Soup is designed for parsing HTML and XML documents, not PDFs.

Option C: unstructured: This Python package is specifically designed to work with unstructured data, including extracting text from PDFs. It provides functionalities to handle various types of content in documents with minimal coding, making it ideal for the task.

Option D: numpy: Numpy is a powerful library for numerical computing in Python and does not provide any tools for text extraction from PDFs.

Given the requirement, Option C (unstructured) is the most appropriate as it directly addresses the need to efficiently extract text from PDF documents with minimal code.

Question 4

Question Type: MultipleChoice

A team wants to serve a code generation model as an assistant for their software developers. It should support multiple programming languages. Quality is the primary objective.

Which of the Databricks Foundation Model APIs, or models available in the Marketplace, would be the best fit?

Options:

A- Llama2-70b

B- BGE-large

C- MPT-7b

D- CodeLlama-34B

Answer:

D

Explanation:

For a code generation model that supports multiple programming languages and where quality is the primary objective, CodeLlama-34B is the most suitable choice. Here's the reasoning:

Specialization in Code Generation: CodeLlama-34B is specifically designed for code generation tasks. This model has been trained with a focus on understanding and generating code, which makes it particularly adept at handling various programming languages and coding contexts.

Capacity and Performance: The '34B' indicates a model size of 34 billion parameters, suggesting a high capacity for handling complex tasks and generating high-quality outputs. The large model size typically correlates with better understanding and generation capabilities in diverse scenarios.

Suitability for Development Teams: Given that the model is optimized for code, it will be able to assist software developers more effectively than general-purpose models. It understands coding syntax, semantics, and the nuances of different programming languages.

Why Other Options Are Less Suitable:

A (Llama2-70b): While also a large model, it's more general-purpose and may not be as fine-tuned for code generation as CodeLlama.

B (BGE-large): This model may not specifically focus on code generation.

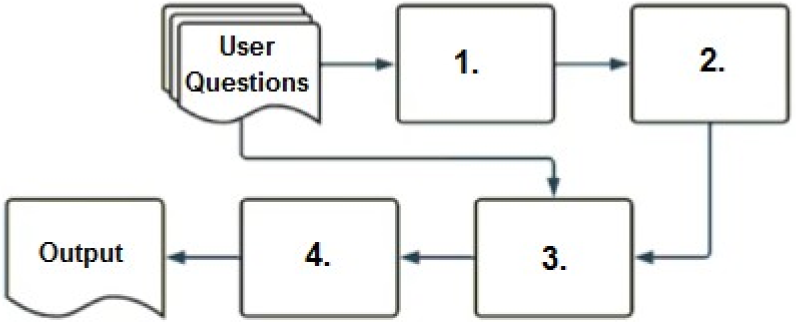
C (MPT-7b): Smaller than CodeLlama-34B and likely less capable in handling complex code generation tasks at high quality.

Therefore, for a high-quality, multi-language code generation application, CodeLlama-34B (option D) is the best fit.

Question 5

Question Type: MultipleChoice

A company has a typical RAG-enabled, customer-facing chatbot on its website.



Select the correct sequence of components a user's questions will go through before the final output is returned. Use the diagram above for reference.

Options:

A- 1.embedding model, 2.vector search, 3.context-augmented prompt, 4.response-generating LLM

B- 1.context-augmented prompt, 2.vector search, 3.embedding model, 4.response-generating LLM

C- 1.response-generating LLM, 2.vector search, 3.context-augmented prompt, 4.embedding model

D- 1.response-generating LLM, 2.context-augmented prompt, 3.vector search, 4.embedding model

Answer:

A

Explanation:

To understand how a typical RAG-enabled customer-facing chatbot processes a user's question, let's go through the correct sequence as depicted in the diagram and explained in option A:

Embedding Model (1): The first step involves the user's question being processed through an embedding model. This model converts the text into a vector format that numerically represents the text. This step is essential for allowing the subsequent vector search to operate effectively.

Vector Search (2): The vectors generated by the embedding model are then used in a vector search mechanism. This search identifies the most relevant documents or previously answered questions that are stored in a vector format in a database.

Context-Augmented Prompt (3): The information retrieved from the vector search is used to create a context-augmented prompt. This step involves enhancing the basic user query with additional relevant information gathered to ensure the generated response is as accurate and informative as possible.

Response-Generating LLM (4): Finally, the context-augmented prompt is fed into a response-generating large language model (LLM). This LLM uses the prompt to generate a coherent and contextually appropriate answer, which is then delivered as the final output to the user.

Why Other Options Are Less Suitable:

B, C, D: These options suggest incorrect sequences that do not align with how a RAG system typically processes queries. They misplace the role of embedding models, vector search, and response generation in an order that would not facilitate effective information retrieval and response generation.

Thus, the correct sequence is embedding model, vector search, context-augmented prompt, response-generating LLM, which is option A.

Question 6

Question Type: **MultipleChoice**

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?

Options:

A- Randomness

B- Diversity of responses

C- Lack of relevance

D- Repetition of responses

Answer:

B

Explanation:

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, and diversity of responses helps 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.

Question 7

Question Type: MultipleChoice

A Generative AI Engineer is creating an LLM-powered application that will need access to up-to-date news articles and stock prices.

The design requires the use of stock prices which are stored in Delta tables and finding the latest relevant news articles by searching the internet.

How should the Generative AI Engineer architect their LLM system?

Options:

A- Use an LLM to summarize the latest news articles and lookup stock tickers from the summaries to find stock prices.

B- Query the Delta table for volatile stock prices and use an LLM to generate a search query to investigate potential causes of the stock volatility.

C- Download and store news articles and stock price information in a vector store. Use a RAG architecture to retrieve and generate at runtime.

D- Create an agent with tools for SQL querying of Delta tables and web searching, provide retrieved values to an LLM for generation of response.

Answer:

D

Explanation:

To build an LLM-powered system that accesses up-to-date news articles and stock prices, the best approach is to create an agent that has access to specific tools (option D).

Agent with SQL and Web Search Capabilities: By using an agent-based architecture, the LLM can interact with external tools. The agent can query Delta tables (for up-to-date stock prices) via SQL and perform web searches to retrieve the latest news articles. This modular approach ensures the system can access both structured (stock prices) and unstructured (news) data sources dynamically.

Why This Approach Works:

SQL Queries for Stock Prices: Delta tables store stock prices, which the agent can query directly for the latest data.

Web Search for News: For news articles, the agent can generate search queries and retrieve the most relevant and recent articles, then pass them to the LLM for processing.

Why Other Options Are Less Suitable:

A (Summarizing News for Stock Prices): This convoluted approach would not ensure accuracy when retrieving stock prices, which are already structured and stored in Delta tables.

B (Stock Price Volatility Queries): While this could retrieve relevant information, it doesn't address how to obtain the most up-to-date news articles.

C (Vector Store): Storing news articles and stock prices in a vector store might not capture the real-time nature of stock data and news updates, as it relies on pre-existing data rather than dynamic querying.

Thus, using an agent with access to both SQL for querying stock prices and web search for retrieving news articles is the best approach for ensuring up-to-date and accurate responses.

Question 8

Question Type: MultipleChoice

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?

Options:

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

Explanation:

The task at hand is to improve the LLM's ability to generate poem-like article summaries with the desired tone and style. Using a neutralizer to 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 a specific 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) is not an effective method for achieving the goal of generating stylized poetic summaries.