EXAMTOPICS

-Expert Verified, Online, **Free**.



CERTIFICATION TEST

-CertificationTest.net - Cheap & Quality Resources With Best Support

Question #1 *Topic 1*

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.

Suggested Answer: *C, E*

Community vote distribution



MOTSOJ Most Recent 7 months ago

Selected Answer: CE

The only options that makes sense

upvoted 2 times

Bh0510 10 months ago

Selected Answer: CE

These are the only options that address chunking evaluation

upvoted 1 times

ee454aa 10 months ago

Selected Answer: CE

C&E is my intuition for this question

upvoted 1 times

bafolmiko 11 months ago

Selected Answer: CE

CE is the correct answer.

IT-Exams-Pro is always the GOAT when it comes to databricks anytime.

upvoted 1 times

kaike_reis 1 year, 7 months ago

Selected Answer: CE

The only options that talk about chunking evaluation.

upvoted 2 times

Retko 1 year, 10 months ago

Selected Answer: CE

Agree with CE

upvoted 1 times

tfaw 1 year, 11 months ago

Selected Answer: CE

I agree with C&E

upvoted 1 times

trendy01 2 years ago

D has limitations depending on the number of tokens in the LLM, which may be disadvantageous in constructing a chunk strategy that sufficiently reflects the context of the question. C and E are more comprehensive and effective optimization strategies because they experimentally evaluate various chunk division strategies and directly measure the suitability of questions and answers.

upvoted 1 times

awron_durat 2 years ago

Answers CE make sense to me! Either using a metric and comparing different methods or LLM as a judge makes sense to me.

upvoted 1 times

Rajmlops 2 years, 1 month ago

right answer

upvoted 1 times

Question #2 *Topic 1*

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

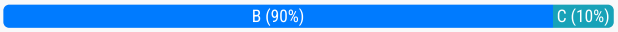
B. Ingest documents from a source –> Index the documents and save to Vector Search –> User submits queries against an LLM –> LLM retrieves relevant documents –> LLM generates a response -> Evaluate model –> Deploy it using Model Serving

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

Suggested Answer: *B*

Community vote distribution



dsnano Most Recent 10 months, 1 week ago
The question asks about build and deploy. B includes the use stage involved. C is incomplete as well, after saving to vector search, maybe it should explicitly mention "Generating response by LLM"?
upvoted 1 times

7e28dac 1 year, 4 months ago
B is correct, as the evaluation happens when the answer is generated
upvoted 1 times

francisrod01 1 year, 4 months ago
The sentence asks for the steps needed to build and deploy the LLM. None of the answers has the correct step-by-step. 1) Build and deploy steps: Ingest documents from a source –> Index the documents and save to Vector Search –> Evaluate model –> Deploy it using Model Serving. 2) usage steps: User submits queries against an LLM –> LLM retrieves relevant documents –> LLM generates a response
upvoted 2 times

Ryan2025 1 year, 6 months ago
B is correct
upvoted 2 times

kaike_reis 1 year, 7 months ago
B is the correct one.
upvoted 2 times

Retko 1 year, 10 months ago
Agree with B
upvoted 1 times

tfaw 1 year, 11 months ago
B in my opinion
upvoted 1 times

awron_durat 2 years ago
First, the RAG application including the LLM has to be built, and then you can evaluate and deploy, so B.
upvoted 3 times

Question #3 *Topic 1*

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

Suggested Answer: *A*

Community vote distribution

A (100%)

sasha_chak Most Recent 10 months ago

Selected Answer: A

B — useful for cost analysis, but not a core production metric.

C — a training metric, not relevant in production.

D — benchmark scores don't reflect real-world customer service performance.

upvoted 2 times

kaike_reis 1 year, 7 months ago

Selected Answer: A

A is the only applicable.

upvoted 1 times

tfaw 1 year, 11 months ago

Selected Answer: A

I agree with A

upvoted 2 times

Arifai900 1 year, 11 months ago

Selected Answer: A

A is the correct answer since evaluating customer service, perplexity score (option C) of the training model does not matter. B and C are irrelevant to the question.

upvoted 1 times

awron_durat 2 years ago

I said A inquiries/time because B and C are more for optimization or training and D would be for the development phase when originally building the application, not production.

upvoted 3 times

Question #4 *Topic 1*

A Generative AI Engineer is building a Generative AI system that suggests the best matched employee team member to newly scoped projects. The team member is selected from a very large team. The match should be based upon project date availability and how well their employee profile matches the project scope. Both the employee profile and project scope are unstructured text.
How should the Generative AI Engineer architect their system?

A. Create a tool for finding available team members given project dates. Embed all project scopes into a vector store, perform a retrieval using team member profiles to find the best team member.

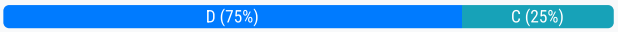
B. Create a tool for finding team member availability given project dates, and another tool that uses an LLM to extract keywords from project scopes. Iterate through available team members' profiles and perform keyword matching to find the best available team member.

C. Create a tool to find available team members given project dates. Create a second tool that can calculate a similarity score for a combination of team member profile and the project scope. Iterate through the team members and rank by best score to select a team member.

D. Create a tool for finding available team members given project dates. Embed team profiles into a vector store and use the project scope and filtering to perform retrieval to find the available best matched team members.

Suggested Answer: *D*

Community vote distribution



omkarg01 9 months, 1 week ago

Selected Answer: D

C is like reinventing the vector search by tool which is not needed so D is right

upvoted 1 times

sasha_chak 10 months ago

Selected Answer: D

For C — Iterating through all team members is inefficient and does not scale for large teams

upvoted 1 times

VittalManikonda 1 year, 2 months ago

Selected Answer: C

C might be the right answer, as the similarity search will give the nearest neighbor, and that is what we need

upvoted 2 times

kaike_reis 1 year, 7 months ago

Selected Answer: D

D is the correct one. C don't mention the embedding method over the text or cite the filtering.

upvoted 1 times

fa2bede 1 year, 11 months ago

Selected Answer: D

Answer D, in the question we are told of "a very large team". Hence, iterating over that list of team members as C suggests does not scale well

upvoted 3 times

trendy01 2 years ago

Answer D,

From a Generative AI engineering perspective, D is the most effective and practical approach for optimizing the match between project scope and team profile.

upvoted 1 times

trendy01 2 years ago

C, This method ranks team members by checking their availability and calculating a similarity score between their profile and project scope, allowing you to recommend the most suitable team members by taking both factors into account. The score-based approach allows you to quantitatively assess the suitability of each team member, contributing to increasing the reliability of recommendations.

upvoted 2 times

awron_durat 2 years ago

A and D were both good answers at first but it makes more sense to embed team profiles than project scopes because of the way you'd want to do searches on the system.The project scope should act as the search query, not as the stored documents. Therefore, Option A reverses the retrieval direction, making the architecture incorrect.

upvoted 1 times

awron_durat 2 years ago

A and D were both good answers at first but it makes more sense to embed team profiles than project scopes because of the way you'd want to do searches on the system.

upvoted 1 times

shashankraj Most Recent 4 months ago

The project scope should act as the search query, not as the stored documents.

Therefore, Option A reverses the retrieval direction, making the architecture incorrect.

upvoted 1 times

Question #5 *Topic 1*

A Generative AI Engineer is designing an LLM-powered live sports commentary platform. The platform provides real-time updates and LLM-generated analyses for any users who would like to have live summaries, rather than reading a series of potentially outdated news articles. Which tool below will give the platform access to real-time data for generating game analyses based on the latest game scores?

A. DatabricksIQ

B. Foundation Model APIs

C. Feature Serving

D. AutoML

Suggested Answer: *C*

Community vote distribution

C (100%)

Arifai900 Most Recent 1 year, 11 months ago

Selected Answer: C

Feature serving is the right answer, to get the live responses one must use feature serving in databricks.

upvoted 2 times

trendy01 2 years ago

Answer C,

B. Foundation Model APIs

Description: An API that accesses the LLM model, allowing you to use the model's features, but is not suitable for collecting real-time sports data directly.

C.Feature Serving

Description: Feature Serving is a function that provides features that a machine learning model can use in real time. This allows real-time sports data (e.g. scores, statistics) to be fed into the model, making it ideal for generating analytics based on this data.

upvoted 4 times

Harry_D 2 years, 1 month ago

I am changing my vote. I think C. Feature Serving is the correct answer. In this post from microsoft, it talks about what is feature serving. https://learn.microsoft.com/en-us/azure/databricks/machine-learning/feature-store/feature-function-serving With Databricks Feature Serving, you can serve structured data for retrieval augmented generation (RAG) applications, as well as features that are required for other applications, such as models served outside of Databricks or any other application that requires features based on data in Unity Catalog. The code provided in https://learn.microsoft.com/en-us/azure/databricks/machine-learning/feature-store/feature-serving-tutorial gives an example of how to use feature serving.I agree! DatabricksIQ and AutoML are more focused on model development and optimization, not real-time data provisioning. Foundation Model APIs provide general pre-trained models for tasks, but don't handle live data integration for near real-time sports commentary.

upvoted 3 times

Harry_D 2 years, 1 month ago

I am changing my vote. I think C. Feature Serving is the correct answer. In this post from microsoft, it talks about what is feature serving. https://learn.microsoft.com/en-us/azure/databricks/machine-learning/feature-store/feature-function-serving

With Databricks Feature Serving, you can serve structured data for retrieval augmented generation (RAG) applications, as well as features that are required for other applications, such as models served outside of Databricks or any other application that requires features based on data in Unity Catalog.

The code provided in https://learn.microsoft.com/en-us/azure/databricks/machine-learning/feature-store/feature-serving-tutorial gives an example of how to use feature serving.

upvoted 3 times

awron_durat 2 years ago

I agree! DatabricksIQ and AutoML are more focused on model development and optimization, not real-time data provisioning. Foundation Model APIs provide general pre-trained models for tasks, but don't handle live data integration for near real-time sports commentary.

upvoted 1 times

Harry_D 2 years, 1 month ago

I vote for B. Foundation Model API.

upvoted 1 times

Question #6 *Topic 1*

A Generative AI Engineer has a provisioned throughput model serving endpoint as part of a RAG application and would like to monitor the serving endpoint's incoming requests and outgoing responses. The current approach is to include a micro-service in between the endpoint and the user interface to write logs to a remote server.
Which Databricks feature should they use instead which will perform the same task?

A. Vector Search

B. Lakeview

C. DBSQL

D. Inference Tables

Suggested Answer: *D*

Community vote distribution

D (100%)

Lonojack Most Recent 1 year, 4 months ago

Selected Answer: D

Inference Tables refer to a built-in Databricks feature that logs and manages the input/output records of machine learning model endpoints. Automatically records incoming requests (like prompts or features) and outgoing responses (like LLM predictions) from a model serving endpoint. Helps trace errors, audit outputs, track usage patterns, and monitor model performance in real time.

upvoted 4 times

Arifai900 1 year, 11 months ago

Selected Answer: D

Inference tables are best suited to log inferencing results and logs into delta tables.

upvoted 1 times

trendy01 2 years ago

D, Inference Tables are used to store and manage prediction results from machine learning models. This provides the ability to record and monitor forecast data.

This is because it is useful for recording request and response data and managing the results. Therefore, it can serve as an effective tool for monitoring incoming requests and outgoing responses in real time.

upvoted 4 times

awron_durat 2 years ago

Must be D. Vector Search makes no sense, DBSQL doesn't relate to log tasks, and Lakeview says "Lakeview Dashboards provide users with a comprehensive toolset for creating, analyzing, and disseminating data-driven insights." from this website:

upvoted 2 times

Question #7 *Topic 1*

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

Suggested Answer: *D*

Community vote distribution

D (100%)

CMD0725 Most Recent 1 year, 3 months ago

Selected Answer: D

Restricing or informing User, or increasing the frequency doesn't eliminate toxicity. Curating upstream data to remove unwanted stuff through manual review will remove toxicity

upvoted 1 times

francisrod01 1 year, 4 months ago

Selected Answer: D

Integrating a manual review process in RAG systems addresses several critical concerns:

- Accuracy: Manual reviews by experts help to verify the correctness of the information retrieved and generated, reducing the chance of inaccuracies or misleading responses often referred to as "hallucinations" in AI.

- Domain Expertise: Trained professionals can provide nuanced insights that automated systems alone cannot achieve, especially in complex fields like healthcare or specialized technical documents.

- Quality Assurance: Manual reviews serve as an audit mechanism, ensuring that the AI-generated responses meet the established standards of quality and reliability, particularly when handling sensitive information.

upvoted 2 times

Arifai900 1 year, 11 months ago

Selected Answer: D

D is the correct answer. Manual review is necessary for a RAG system with irregular responses.

upvoted 2 times

Question #8 *Topic 1*

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

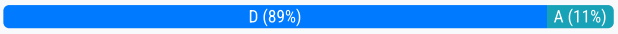
B. context length 2048: smallest model is 11GB and embedding dimension 2560

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

Suggested Answer: *D*

Community vote distribution



Qix Highly Voted 1 year, 9 months ago

D is the correct solution, as the example present in the official Databricks exam guide

upvoted 5 times

sasha_chak Most Recent 10 months ago

Options B and C propose significantly larger models (11GB and 14GB) and embedding dimensions, which would drastically increase both latency and operational costs, directly contradicting the requirement. Option A is viable, but since D is smaller (0.13GB vs 0.44GB) and still fits the token limit, D is the better choice for optimizing speed and cost.

upvoted 1 times

6da92db 1 year, 6 months ago

A. You want the smallest model that can accommodate 512 tokens in the context window and VS input token size.

upvoted 1 times

Arifai900 1 year, 11 months ago

D is correct.

upvoted 1 times

awron_durat 2 years ago

D is correct.

upvoted 1 times

srihdar 2 years, 1 month ago

D is the answer Since cost and latency are more important than quality for this application, the smallest model with the shortest context length (512 tokens) will be the most efficient in terms of resource usage and speed. This option provides lower latency and cost compared to the larger models, making it a suitable choice for the engineer's requirements.

upvoted 3 times

Question #9 *Topic 1*

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

Suggested Answer: *B*

Community vote distribution

B (100%)

e7ec3f6 Most Recent 1 year ago

Selected Answer: B

A domain-specific LLM (e.g., BioGPT, Med-PaLM-small, Clinical-T5) is pre-trained on biomedical literature → better semantic understanding, lower hallucination, more accurate answers on cancer research queries

upvoted 1 times

Vas1l1k0s 1 year, 1 month ago

Selected Answer: B

The correct answer is B. Pick a smaller LLM that is domain-specific.

For a cost-conscious startup in cancer research, the goal is to balance quality, domain relevance, and affordability. Here's why a smaller, domain-specific LLM is ideal:

Lower cost: Smaller models are cheaper to run and faster to respond.

Higher relevance: Domain-specific models (e.g., biomedical or clinical LLMs) are trained on specialized data, making them more accurate for cancer-related queries.

Efficient retrieval: When paired with a well-curated retriever, even a smaller model can generate high-quality answers.

This strategy ensures the startup delivers meaningful results without overspending on compute or API usage.

upvoted 1 times

Manish_Kum 1 year, 8 months ago

Selected Answer: B

For a small and cost-conscious startup in the cancer research field, the best strategy to build a good-quality Retrieval-Augmented Generation (RAG) application while being cost-conscious and able to cater to customer needs would be - B. Pick a smaller LLM that is domain-specific

upvoted 2 times

Question #10 *Topic 1*

A Generative AI Engineer is responsible for developing a chatbot to enable their company's internal HelpDesk Call Center team to more quickly find related tickets and provide resolution. While creating the GenAI application work breakdown tasks for this project, they realize they need to start planning which data sources (either Unity Catalog volume or Delta table) they could choose for this application. They have collected several candidate data sources for consideration: call_rep_history: a Delta table with primary keys representative_id, call_id. This table is maintained to calculate representatives' call resolution from fields call_duration and call start_time. transcript Volume: a Unity Catalog Volume of all recordings as a *.wav files, but also a text transcript as *.txt files. call_cust_history: a Delta table with primary keys customer_id, cal1_id. This table is maintained to calculate how much internal customers use the HelpDesk to make sure that the charge back model is consistent with actual service use. call_detail: a Delta table that includes a snapshot of all call details updated hourly. It includes root_cause and resolution fields, but those fields may be empty for calls that are still active. maintenance_schedule – a Delta table that includes a listing of both HelpDesk application outages as well as planned upcoming maintenance downtimes.
They need sources that could add context to best identify ticket root cause and resolution.
Which TWO sources do that? (Choose two.)

A. call_cust_history

B. maintenance_schedule

C. call_rep_history

D. call_detail

E. transcript Volume

Suggested Answer: *D, E*

Community vote distribution

DE (100%)

sasha_chak 9 months, 2 weeks ago

Selected Answer: DE

call_rep_history – tracks how long calls take, not how to fix them. call_cust_history – focuses on billing and usage stats, which don't provide technical answers. maintenance_schedule – shows system status (outages), but doesn't help troubleshoot specific user errors or past tickets.

upvoted 1 times

djohn_prasad Most Recent 9 months, 1 week ago

Call rep history is option C not D

upvoted 1 times

trendy01 2 years ago

D.call_detail This delta table contains the details of the call and has root_cause and resolution fields. This provides important information to understand and solve the problem.

E. Transcript Volume This Unity catalog volume includes text transcriptions of conversations with customers. Provide specific details related to your customer's issue to help determine the root cause.

upvoted 1 times

HemaKG 2 years, 1 month ago

I got answer as B & E. Request to review into the same.

upvoted 1 times

Question #11 *Topic 1*

What is the most suitable library for building a multi-step LLM-based workflow?

A. Pandas

B. TensorFlow

C. PySpark

D. LangChain

Suggested Answer: *D*

Community vote distribution

D (100%)

tenoke7576 Most Recent 1 month ago

Selected Answer: D

Langchain is the only LLM library from the list

upvoted 1 times

apimenta 1 year, 3 months ago

Selected Answer: D

D, no doubt

upvoted 4 times

DalyBread 1 year, 3 months ago

Selected Answer: D

Langchain referenced in databricks docs

upvoted 2 times

7e28dac 1 year, 4 months ago

Selected Answer: D

The correct answer should be Langchain, which is the most commonly used framework to integrate LLM Models

upvoted 1 times

seaun 1 year, 5 months ago

Selected Answer: D

Answer D LangChain

LangChain is an open-source framework specifically designed for building applications powered by large language models (LLMs). It provides tools and components that make it easy to chain multiple prompts, integrate LLMs with external data sources, manage memory, and create multi-step workflows. LangChain is known for its ability to create AI agents that can handle complex tasks by combining LLMs with various tools and APIs

upvoted 2 times

Question #12 *Topic 1*

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.

Suggested Answer: *B*

Community vote distribution

D (100%)

fa2bede Most Recent 1 year, 11 months ago

Selected Answer: D

The answer must be D. You should not reach out after you started using

upvoted 2 times

HardLearner 2 years ago

Selected Answer: D

Definitely D, it is not appropriate to use model either in dev or prod before checking the licensing

upvoted 2 times

568f95c 2 years, 1 month ago

Selected Answer: D

You can use personal data because it has not licensing if it is yours, You should not reach out after using data from external source, to find out if you now can use it.

upvoted 4 times

mojj 2 years, 1 month ago

The answer must be D

upvoted 3 times

Question #13 *Topic 1*

A Generative AI Engineer is testing a simple prompt template in LangChain using the code below, but is getting an error.

Assuming the API key was properly defined, what change does the Generative AI Engineer need to make to fix their chain?

```
from langchain.chains import LLMChain
from langchain_community.llms import OpenAI
from langchain_core.prompts import PromptTemplate

prompt_template = "Tell me a {adjective} joke"

prompt = PromptTemplate(
     input_variables=["adjective"],
     template=prompt_template
)

llm = LLMChain(prompt=prompt)
llm.generate([{"adjective": "funny"}])
```

```
prompt_template = "Tell me a {adjective} joke"

prompt = PromptTemplate(
     input_variables=["adjective"],
     template=prompt_template
)

llm = LLMChain(prompt=prompt)
llm.generate("funny")
```

```
prompt_template = "Tell me a {adjective} joke"

prompt = PromptTemplate(
     input_variables=["adjective"],
     template=prompt_template
)

llm = LLMChain(prompt=prompt.format("funny"))
llm.generate()
```

```
prompt_template = "Tell me a {adjective} joke"

prompt = PromptTemplate(
     input_variables=["adjective"],
     template=prompt_template
     llm=OpenAI()
)

llm = LLMChain(prompt=prompt)
llm.generate([{"adjective": "funny"}])
```

```
prompt_template = "Tell me a {adjective} joke"

prompt = PromptTemplate(
     input_variables=["adjective"],
     template=prompt_template
)

llm = LLMChain(llm=OpenAI(), prompt=prompt)
llm.generate([{"adjective": "funny"}])
```

A.

B.

C.

D.

Suggested Answer: *C*

Community vote distribution

D (100%)

7e28dac Most Recent 1 year, 4 months ago

Selected Answer: D

D is the right answer, as you would need llm and prompt defined in LLM Chain

upvoted 1 times

tfaw 1 year, 11 months ago

Selected Answer: D

D is the answer

upvoted 1 times

fr2022 2 years ago

Selected Answer: D

from langchain.chains import LLMChain from langchain_community.llms import OpenAI from langchain_core.prompts import PromptTemplate prompt_template = "Tell me a {adjective} joke" prompt = PromptTemplate(input_variables=["adjective"], template=prompt_template) llm = LLMChain(llm=OpenAI(), prompt=prompt)

upvoted 1 times

4af18fc 2 years, 1 month ago

Selected Answer: D

Option D is correct as we need to pass the OpenAi() constructor to the LLMChain()

upvoted 3 times

Question #14 *Topic 1*

A Generative AI Engineer is creating an LLM system that will retrieve news articles from the year 1918 and related to a user's query and summarize them. The engineer has noticed that the summaries are generated well but often also include an explanation of how the summary was generated, which is undesirable.
Which change could the Generative AI Engineer perform to mitigate this issue?

A. Split the LLM output by newline characters to truncate away the summarization explanation.

B. Tune the chunk size of news articles or experiment with different embedding models.

C. Revisit their document ingestion logic, ensuring that the news articles are being ingested properly.

D. Provide few shot examples of desired output format to the system and/or user prompt.

Suggested Answer: *D*

Community vote distribution

D (100%)

vish9 Most Recent 10 months ago

Yes I agree with D.

upvoted 1 times

7e28dac 1 year, 4 months ago

The summaries are already generated very well. So a few shots would do the trick

upvoted 1 times

seaun 1 year, 5 months ago

This technique involves providing the LLM with a few example input-output pairs to guide its behavior and demonstrate the desired format. By showing the model how you want the output to look (e.g., just the summary without any explanation), you significantly increase the chances that it will adhere to that format in subsequent responses.

upvoted 2 times

Question #15 *Topic 1*

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

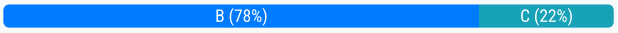
B. Fine-tune the model on your data, hoping it will learn what is appropriate and not

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.

Suggested Answer: *B*

Community vote distribution



Tiger4Code Most Recent 3 weeks ago

Selected Answer: B

b: hope does not guarantee anything

upvoted 1 times

Bh0510 10 months ago

Selected Answer: B

B is the correct answer. "Hoping" is a giveaway.

upvoted 1 times

seaun 1 year, 5 months ago

Selected Answer: B

B is correct

upvoted 1 times

adaine 1 year, 6 months ago

Selected Answer: B

B is correct

upvoted 1 times

dementor 1 year, 10 months ago

Selected Answer: B

B is the correct answer.

upvoted 2 times

fa2bede 1 year, 11 months ago

Selected Answer: C

The correct answer should be C. Limiting the data available may not help solve the hallucination problem. Fine-tuning helps reduce hallucination

upvoted 2 times

fa2bede 1 year, 11 months ago

The correct answer should be C. Limiting the data available may not help solve the hallucination problem. Fine-tuning helps reduce hallucinationI change my mind, B is correct. Fine tuning in this use case can be very likely increase hallucination

upvoted 2 times

Crunch 1 year, 10 months ago

Selected Answer: B

B. Hoping for the best is not the way, the fine tuned data can still contain sensitive information

upvoted 2 times

fa2bede 1 year, 11 months ago

I change my mind, B is correct. Fine tuning in this use case can be very likely increase hallucination

upvoted 1 times

Question #16 *Topic 1*

A Generative Al Engineer interfaces with an LLM with prompt/response behavior that has been trained on customer calls inquiring about product availability. The LLM is designed to output “In Stock” if the product is available or only the term “Out of Stock” if not.
Which prompt will work to allow the engineer to respond to call classification labels correctly?

A. Respond with “In Stock” if the customer asks for a product.

B. You will be given a customer call transcript where the customer asks about product availability. The outputs are either “In Stock” or “Out of Stock”. Format the output in JSON, for example: {“call_id”: “123”, “label”: “In Stock”}.

C. Respond with “Out of Stock” if the customer asks for a product.

D. You will be given a customer call transcript where the customer inquires about product availability. Respond with “In Stock” if the product is available or “Out of Stock” if not.

Suggested Answer: *B*

Community vote distribution



jn_uk Most Recent 1 year ago

Selected Answer: D

B seems helpful at first glance because it's talking about JSON. However.... there is nothing in the question that *asks* for JSON, so this is a red herring.

D includes a description for when to use each label. So D is the answer.

upvoted 1 times

ConquerorAlpha 1 year, 10 months ago

Selected Answer: D

D should be the correct answer because both the options are provided in this option only.

upvoted 2 times

norbertpolcz 1 year, 10 months ago

Selected Answer: D

I think it's clearly option D. option B might seem right at first but there are two issues. 1. It does not specify when to use "In stock" and "Out of stock" in the response. We might can presume that the model already understands when to use which but then 2. The question says that the output contains only "In stock" or "Out of stock" and option B has a different output format.

upvoted 4 times

trendy01 2 years ago

Selected Answer: B

B is providing a **specific format (JSON format)** for engineers to classify inventory status for customer inquiries via LLM. This requires the model to clearly output inventory status and output this structured in JSON format, making it suitable for post-processing the data or using it in automated systems.

Compared to other options, B provides specific guidelines to accurately manage classification labels.

A and C each require an “In Stock” or “Out of Stock” response, but do not provide clear guidance on specific labeling formats or processes.

D requires a response about inventory status, but is insufficient because it does not provide a specific data format or structured output.

upvoted 2 times

maciejmirski 2 years ago

Selected Answer: D

D is correct

upvoted 3 times

HemaKG 2 years, 1 month ago

D. You will be given a customer call transcript where the customer inquiries about product availability. Respond with “In Stock” if the product is available or “Out of Stock” if not.

Dear service provider, Please do not leave it with CORRECT ANSWER and Community Vote Distribution. Please provide silid explanation.

upvoted 3 times

4af18fc 2 years, 1 month ago

Selected Answer: D

Option B doesn't tell when to use "In Stock" and "Out of Stock".

D should be the correct answer

upvoted 4 times

Question #17 *Topic 1*

A Generative AI Engineer is tasked with developing a RAG application that will help a small internal group of experts at their company answer specific questions, augmented by an internal knowledge base. They want the best possible quality in the answers, and neither latency nor throughput is a huge concern given that the user group is small and they're willing to wait for the best answer. The topics are sensitive in nature and the data is highly confidential and so, due to regulatory requirements, none of the information is allowed to be transmitted to third parties. Which model meets all the Generative AI Engineer's needs in this situation?

A. Dolly 1.5B

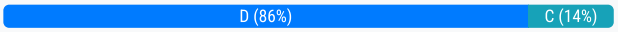
B. OpenAI GPT-4

C. BGE-large

D. Llama2-70B

Suggested Answer: *D*

Community vote distribution



Tiger4Code Most Recent 3 weeks ago

Selected Answer: D

Correct answer: D. Llama 2 70B

Why?

- High answer quality: 70B parameters make it the strongest general-purpose model among the listed open-weight options.

- Data confidentiality: It can be deployed on the company's own infrastructure, keeping confidential data internal.

- Latency is not a concern: The company can prioritize answer quality over speed.

Why not the others?

- A. Dolly 1.5B: Too small for the best answer quality.

- B. OpenAI GPT-4: A hosted API may transmit data to a third party, conflicting with the requirement.

- C. BGE-large: Primarily an embedding model for retrieval, not a generative answer model.

upvoted 1 times

7e28dac 1 year, 4 months ago

Selected Answer: D

BGE Large is an embedding based LLM model (embeddings available to serving endpoints). and it is ideal for retrieval step, as mentioned in the comments

upvoted 1 times

DavidMiller 1 year, 6 months ago

Selected Answer: D

While BGE-large could be part of the RAG system for the retrieval step (embedding the knowledge base and queries for vector search), the question asks for a single model to meet all needs, implying the generative component of RAG. BGE-large alone cannot fulfill the generation requirement, whereas Llama2-70B can handle generation and be paired with an embedding model if needed.

upvoted 1 times

DavidMiller 1 year, 6 months ago

Selected Answer: D

GE-large can’t generate answers, failing to meet this core need. Additionally, the focus on “best possible quality” and handling sensitive data suggests a generative model that can be deployed on-premises, which BGE-large doesn’t address as an embedding-focused model.

upvoted 1 times

jothi28021951 1 year, 6 months ago

Selected Answer: D

The engineer needs "the best possible quality in the answers." Llama2-70B is one of the largest and most capable open-source language models available, known for producing high-quality outputs. So the answer is Llama2-70B model

upvoted 1 times

adaine 1 year, 6 months ago

Selected Answer: D

D is the correct answer.

- A) Dolly is too small so it won't be the best quality.

- B) OpenAI GPT-4 is high quality but there would be issues around confidentiality

- C) BGE-large is an embedding model so it's not suitable for answering questions.

Llama2-70B is sufficiently big to provide high quality answers. Even though it would be slower than smaller models the question says that the users don't mind about latency. It can also be self-hosted so there is no third-party data transmission.

upvoted 2 times

AtulYadav 1 year, 6 months ago

Selected Answer: C

The correct answer is BGE-large which is C.

upvoted 1 times

Question #18 *Topic 1*

A Generative AI Engineer would like an LLM to generate formatted JSON from emails. This will require parsing and extracting the following information: order ID, date, and sender email. Here's a sample email:

They will need to write a prompt that will extract the relevant information in JSON format with the highest level of output accuracy.
Which prompt will do that?

```
Date: April 23, 2024
Time: 4:22 PM
From: anjali.thayer@computex.org
To: cust_service@realtek.com
Subject: Shipment details

Hey there,

I have a shipment (order ID is CD34RFT) can you please send me an update?

Thank you,
Anjali
```

A. You will receive customer emails and need to extract date, sender email, and order ID. You should return the date, sender email, and order ID information in JSON format.

B. You will receive customer emails and need to extract date, sender email, and order ID. Return the extracted information in JSON format. Here's an example: {"date": "April 16, 2024", "sender_email": "sarah.lee925@gmail.com", "order_id": "RE987D"}

C. You will receive customer emails and need to extract date, sender email, and order ID. Return the extracted information in a human-readable format.

D. You will receive customer emails and need to extract date, sender email, and order IReturn the extracted information in JSON format.

Suggested Answer: *B*

Community vote distribution

B (100%)

Vas1l1k0s Most Recent 1 year, 1 month ago

B. Output will be more concise, since prompt has an example.

upvoted 1 times

ellessayy 1 year, 2 months ago

B includes the example format as well

upvoted 1 times

DalyBread 1 year, 3 months ago

B because ot also includes a good example output

upvoted 3 times

Question #19 *Topic 1*

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.

Suggested Answer: *A*

Community vote distribution

A (100%)

Tiger4Code Most Recent 3 weeks ago

Selected Answer: A

a: retereive and pass to prompt to provide up to date context

upvoted 1 times

trendy01 2 years ago

Selected Answer: A

A works by using a retriever to retrieve information from a new document and then inserting it into the prompt to pass it on to the LLM. This structure is effective in providing up-to-date information, reflecting frequently updated documentation, while minimizing cost and development effort.

B describes the update frequency of LLM, but lacks a description of the actual application architecture.

C only mentions prompt engineering and LLM, and doesn't explain how to handle updates for new documents.

D describes an agent-using approach, but lacks specifics on how an agent-using structure would achieve the same effective information retrieval and insertion as A.

upvoted 2 times

Question #20 *Topic 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.

Suggested Answer: *B*

Community vote distribution

B (100%)

richjacome Most Recent 1 month ago

Selected Answer: B

The answer is B.

This describes the standard agent architecture — define the available tools (text retrieval, event date API, standings table query) in a system prompt, and let the agent reason about which tool(s) to call for each query. This is exactly how frameworks like the OpenAI Agents SDK work, as covered in this course.

upvoted 1 times

sasha_chak 9 months, 2 weeks ago

Selected Answer: B

In an agentic workflow, you provide the LLM with a "toolbox" in the system prompt (e.g., "You have access to get_schedule(), query_standings(), and search_docs()")

upvoted 1 times

Maddykrish1984 1 year, 2 months ago

Selected Answer: B

B is the correct answer

upvoted 1 times

Question #21 *Topic 1*

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.

Suggested Answer: *D*

Community vote distribution

D (100%)

richjacome Most Recent 1 month ago

Selected Answer: D

The answer is D.

This describes the standard RAG pipeline for document-based Q&A:

Chunk HR PDFs into manageable pieces

Embed chunks into a vector store

Retrieve the most relevant chunks using the employee's question

Generate a response with an LLM grounded in the retrieved context

upvoted 1 times

6da92db 1 year, 6 months ago

Selected Answer: D

D is the simplest high level architecture for this use-case. However, C is likely great option for improving the model once the there are vetted body of historical Q&As.

upvoted 1 times

Question #22 *Topic 1*

Generative AI Engineer at an electronics company just deployed a RAG application for customers to ask questions about products that the company carries. However, they received feedback that the RAG response often returns information about an irrelevant product.
What can the engineer do to improve the relevance of the RAG's response?

A. Assess the quality of the retrieved context

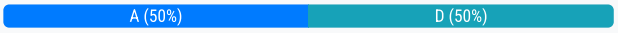
B. Implement caching for frequently asked questions

C. Use a different LLM to improve the generated response

D. Use a different semantic similarity search algorithm

Suggested Answer: *A*

Community vote distribution



81a856e Highly Voted 1 year, 9 months ago

Selected Answer: D

The issue described is that the RAG (Retrieval-Augmented Generation) application retrieves irrelevant context, which indicates that the retrieval mechanism is not accurately identifying the most relevant information. By using a different semantic similarity search algorithm, the engineer can improve the retrieval step to ensure that the most contextually relevant documents or data are selected before passing them to the LLM for generating responses. Options like A (assessing quality) may help diagnose the issue but do not directly fix it. B (caching) is not related to improving relevance, and C (using a different LLM) addresses generation quality, not retrieval accuracy.

upvoted 5 times

81a856e Highly Voted 1 year, 9 months ago

The issue described is that the RAG (Retrieval-Augmented Generation) application retrieves irrelevant context, which indicates that the retrieval mechanism is not accurately identifying the most relevant information. By using a different semantic similarity search algorithm, the engineer can improve the retrieval step to ensure that the most contextually relevant documents or data are selected before passing them to the LLM for generating responses. Options like A (assessing quality) may help diagnose the issue but do not directly fix it. B (caching) is not related to improving relevance, and C (using a different LLM) addresses generation quality, not retrieval accuracy.D can be second step once quality can be assessed. So, it will be A.

upvoted 5 times

Lg22 1 year, 6 months ago

D can be second step once quality can be assessed. So, it will be A.

upvoted 4 times

Carkeys Most Recent 3 months, 3 weeks ago

"The RAG response often returns information about an irrelevant product", points to the retrieval stage of the RAG pipeline. If the retriever fetches documents about Product B instead of Product A, the LLM is doing exactly what it's supposed to do. "D" changing similarity search algorithm could improve retrieval. The engineer should first assess whether retrieval is actually the problem, in engineering, diagnosis comes before changing the implementation.

upvoted 1 times

saha3333 5 months, 4 weeks ago

Assessing the quality of retrieved context means investigating things like embedding, chunking strategy, similarity algo, filtering. ´D´is not complete because "assessing the retrieved context" is the correct first step, which might lead you to change the algorithm — but jumping straight to swapping algorithms without diagnosing first is premature

upvoted 1 times

krishtgkkinfo 6 months, 3 weeks ago

First Diagnoseit. improve chunking strategy , check if embeddings are poor and then go for a different algo

upvoted 1 times

vn327 8 months, 2 weeks ago

Wrong RAG output → diagnose retrieval first. Assess what's being retrieved before changing anything

upvoted 2 times

vish9 10 months ago

A first see if we are getting correct information to answer the queries from.

upvoted 1 times

DavidMiller 1 year, 6 months ago

A is a useful step for diagnosis but doesn't fix the problem as efficiently as D. D directly improves retrieval by changing the semantic similarity algorithm. Alternatively, use "Semantic Switch Solves" to recall that switching the algorithm (D) solves the relevance issue.

upvoted 1 times

Question #23 *Topic 1*

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

Suggested Answer: *D*

Community vote distribution



mojj Highly Voted 2 years, 1 month ago

A. Safety Guardrail Reasoning: A Safety Guardrail is designed to ensure that a conversational AI system like a chatbot stays within the intended boundaries of its domain, preventing it from generating unsafe or irrelevant responses. In this case, it ensures the chatbot avoids responding to political questions, which is essential for maintaining focus on insurance-related queries and complying with company policies.

upvoted 10 times

Carkeys Most Recent 3 months, 3 weeks ago

Keeping the conversation to insurance is about maintaining context. I wonder if the part of the community voting for "safety" is too focused on the course material emphasizing safety guardrails. If political discussions were a safety issue the question should provide context the user is from a country or region where it is illegal to discuss it.

upvoted 1 times

akc12345789 6 months ago

Why not the others? A. Safety Guardrail Used for harmful, unsafe, violent, self-harm, hate, or toxic content. B. Security Guardrail Used for protecting data, preventing prompt injection, unauthorized access, secrets leakage, etc. D. Compliance Guardrail Used for regulatory/legal policy compliance, such as GDPR, HIPAA, financial rules, privacy rules, audit requirements, etc. Here, the issue is not mainly safety, security, or legal compliance. It is about keeping the chatbot focused on insurance only. So the answer is: C. Contextual Guardrail

upvoted 1 times

erumi 6 months, 1 week ago

It's about the context NOT the safety.

upvoted 1 times

Harsaas 7 months ago

Since its related to context it is C

upvoted 1 times

Lg22 1 year, 6 months ago

Selected Answer: D

It should be D

upvoted 2 times

Sudipta3009 8 months ago

A Safety Guardrail is used to control and restrict model outputs based on predefined rules or policies, such as blocking harmful, sensitive, or out-of-scope content. In this case, preventing the chatbot from answering political questions and returning a fixed response is a classic example of enforcing safe and controlled behavior.

upvoted 1 times

vish9 10 months ago

Contextual guard rail. I checked this answer with chatgpt and gemini

upvoted 2 times

adaine 1 year, 6 months ago

Selected Answer: C

I really think the answer is C. The chatbot is limited to answering only insurance-related questions and must avoid political topics. That's a clear case of keeping the model within a defined scope—which is exactly what contextual guardrails are for.

upvoted 1 times

e7ec3f6 1 year ago

Safety guardrails filter input prompts and block or redirect queries outside the allowed domain

upvoted 1 times

adaine 1 year, 5 months ago

I am conviced by copilot's explanation: In your scenario, the chatbot is not just trying to stay on topic—it is explicitly forbidden from answering a specific category of questions (politics), regardless of the context or flow of conversation. This is a hard restriction, not a matter of staying contextually relevant. That kind of restriction is a hallmark of a Safety Guardrail, which is designed to: Block certain topics (like politics, violence, or adult content), Prevent harmful or inappropriate outputs, Enforce ethical or policy-based boundaries.

upvoted 2 times

Lg22 1 year, 6 months ago

It should be DChanging it to A

upvoted 2 times

Lg22 1 year, 5 months ago

Changing it to A

upvoted 1 times

natadatabricksadf 1 year, 9 months ago

Selected Answer: C

C Contextual Guardrail is used to ensure that the chatbot stays within its intended domain (in this case, insurance) and does not respond to off-topic queries, such as politics

upvoted 1 times

adaine 1 year, 6 months ago

I really think the answer is C. The chatbot is limited to answering only insurance-related questions and must avoid political topics. That's a clear case of keeping the model within a defined scope—which is exactly what contextual guardrails are for.I feel like safety guardrails are more to do with filtering offensive content, e.g. swearing, violent language, etc. rather than about staying on topic.

upvoted 1 times

adaine 1 year, 6 months ago

I feel like safety guardrails are more to do with filtering offensive content, e.g. swearing, violent language, etc. rather than about staying on topic.

upvoted 2 times

jothi28021951 1 year, 6 months ago

Safety Guardrails in the context of LLMs are designed to prevent the model from generating harmful, inappropriate, or off-topic content. This includes topics that are explicitly disallowed by policy, such as political discussions in an insurance chatbot. The goal is to ensure the chatbot operates within acceptable boundaries and doesn't produce undesirable outputs.

upvoted 1 times

CoolSmartDude 1 year, 7 months ago

Safey Guardrail

upvoted 1 times

natadatabricksadf 1 year, 9 months ago

C Contextual Guardrail is used to ensure that the chatbot stays within its intended domain (in this case, insurance) and does not respond to off-topic queries, such as politicsI've changed my mind correct answer is A. Safety Guardrail

upvoted 1 times

natadatabricksadf 1 year, 9 months ago

I've changed my mind correct answer is A. Safety Guardrail

upvoted 1 times

Soumak 1 year, 10 months ago

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.

upvoted 1 times

trendy01 2 years ago

D. Compliance Guardrail is a suitable choice to solve this problem. Compliance guardrails are a way to ensure that your chatbot adheres to certain policies, where company policy requires it to block answers to political questions and return a standard message.

upvoted 4 times

Question #24 *Topic 1*

A Generative AI Engineer I using the code below to test setting up a vector store:

Assuming they intend to use Databricks managed embeddings with the default embedding model, what should be the next logical function call?

```
from databricks.vector_search.client import VectorSearchClient

vsc = VectorSearchClient()

vsc.create_endpoint(
     name= "vector_search_test",
     endpoint_type= "STANDARD"
)
```

A. vsc.get_index()

B. vsc.create_delta_sync_index()

C. vsc.create_direct_access_index()

D. vsc.similarity_search()

Suggested Answer: *B*

Community vote distribution



itsrafa Most Recent 1 year, 2 months ago

Selected Answer: B

ChatGPT 5 swears it's B(create_delta_sync_index() since the nuance is in the next statement.

upvoted 1 times

DavidMiller 1 year, 6 months ago

Selected Answer: C

As you noted, vsc.create_delta_sync_index() is more suited for production workflows where a Delta table is already in use, and you need the index to automatically sync with it. That's not the case for a minimal test setup. Correct answer: C. vsc.create_direct_access_index()

upvoted 1 times

adaine 1 year, 6 months ago

Selected Answer: C

The correct answer is C. vsc.create_direct_access_index().

This is the appropriate next step when testing the setup of a vector store using Databricks managed embeddings and the default embedding model, especially if you're not using a Delta table. create_direct_access_index() lets you manually add documents and embeddings, which is ideal for testing and minimal setup.

Option B (create_delta_sync_index()) is used when you already have a Delta table and want your index to automatically sync with it — a better fit for production workflows, not initial testing.

upvoted 2 times

Soumak 1 year, 10 months ago

Selected Answer: B

After setting up a vector store endpoint, creating an index is necessary to start populating and organizing the data. The create_delta_sync_index() function specifically creates an index that synchronizes with a Delta table, allowing automatic updates as the data changes. This is likely the most appropriate choice if the engineer plans to use dynamic data that is updated over time.

upvoted 1 times

trendy01 2 years ago

Selected Answer: B

If the question assumes "if you want to use the default embedding model managed by Databricks", then B. vsc.create_delta_sync_index() might be a more appropriate answer.

upvoted 3 times

Question #25 *Topic 1*

A Generative AI Engineer is tasked with deploying an application that takes advantage of a custom MLflow Pyfunc model to return some interim results.
How should they configure the endpoint to pass the secrets and credentials?

A. Use spark.conf.set ()

B. Pass variables using the Databricks Feature Store API

C. Add credentials using environment variables

D. Pass the secrets in plain text

Suggested Answer: *C*

Community vote distribution

C (100%)

adaine Most Recent 1 year, 6 months ago

Selected Answer: C

-C) Environment variables allow you to securely pass credentials (like API keys, database passwords, etc.) to your application without hardcoding them into your code.

upvoted 2 times

Question #26 *Topic 1*

A Generative AI Engineer wants to build an LLM-based solution to help a restaurant improve its online customer experience with bookings by automatically handling common customer inquiries. The goal of the solution is to minimize escalations to human intervention and phone calls while maintaining a personalized interaction. To design the solution, the Generative AI Engineer needs to define the input data to the LLM and the task it should perform.
Which input/output pair will support their goal?

A. Input: Online chat logs; Output: Group the chat logs by users, followed by summarizing each user's interactions

B. Input: Online chat logs; Output: Buttons that represent choices for booking details

C. Input: Customer reviews; Output: Classify review sentiment

D. Input: Online chat logs; Output: Cancellation options

Suggested Answer: *B*

Community vote distribution



juanlu46 Most Recent 2 weeks ago

Selected Answer: A

Talks about inpupt data and task that LLM need to do. Generating buttons with booking details it's not a personalized option. The LLM need to summerize the chat history to personalized the answer.

upvoted 1 times

richjacome 1 month ago

Selected Answer: B

The answer is B — Input: Online chat logs; Output: Buttons that represent choices for booking details.

This directly addresses every stated goal:

Handles common booking inquiries automatically — The LLM reads the customer's chat and responds with structured booking options (date, time, party size, etc.), resolving the inquiry without human involvement.

Minimizes escalation — Structured button choices guide the customer through a self-service booking flow, reducing the need for phone calls or human agents.

Maintains personalized interaction — The LLM interprets the customer's specific chat messages and presents contextually relevant options

upvoted 1 times

Dsrqwert123 1 month, 1 week ago

Selected Answer: A

A since we personize

upvoted 1 times

nedlo 9 months, 3 weeks ago

Selected Answer: A

I changed my decision. I think its A, because key sentence is "while maintaining a personalized interaction", so group by users fits to this.

upvoted 2 times

nedlo 9 months, 3 weeks ago

Selected Answer: B

i would say B. Question does not say anything about cancelation, and no A because we dont need group by user, if customer can be a new user as well

upvoted 1 times

vce2020 10 months, 3 weeks ago

Selected Answer: A

The summary will allow the LLM to understand the provided options and learn better.

upvoted 1 times

kirangud77 11 months, 4 weeks ago

Selected Answer: A

Minimize the escalations - Answer should be A

upvoted 1 times

7e28dac 1 year, 3 months ago

Selected Answer: D

Most reasonable option is cancellations. it should be D

upvoted 1 times

seaun 1 year, 5 months ago

Selected Answer: B

Answer should be B

upvoted 3 times

adaine 1 year, 5 months ago

Selected Answer: A

A, 'summarise the user's interaction' seems the most reasonable choice for 'maintain personalised interaction'.

upvoted 2 times

Lg22 1 year, 6 months ago

Selected Answer: D

It should be D

upvoted 2 times

trendy01 2 years ago

Selected Answer: B

B, This option can be useful for automatically handling customer inquiries in a structured way that allows you to provide immediate responses when customers ask questions about their reservations.

upvoted 3 times

Question #27 *Topic 1*

What is an effective method to preprocess prompts using custom code before sending them to an LLM?

A. Directly modify the LLM's internal architecture to include preprocessing steps

B. It is better not to introduce custom code to preprocess prompts as the LLM has not been trained with examples of the preprocessed prompts

C. Rather than preprocessing prompts, it's more effective to postprocess the LLM outputs to align the outputs to desired outcomes

D. Write a MLflow PyFunc model that has a separate function to process the prompts

Suggested Answer: *D*

Community vote distribution

D (100%)

7e28dac Most Recent 1 year, 3 months ago

Selected Answer: D

D is the most appropriate answer. MLFLow Pyfunc can handle custom functions related to input feed from Delta lake, Prompt Template for LLM's, They are also useful to register the model

upvoted 1 times

trendy01 2 years ago

Selected Answer: D

D appears to be the most appropriate choice. By writing MLflow PyFunc models and having a separate function to handle prompts, you can perform preprocessing more systematically and flexibly. This is a way to improve performance by providing optimized prompts for LLM.

upvoted 3 times

Question #28 *Topic 1*

A Generative AI Engineer is developing an LLM application that users can use to generate personalized birthday poems based on their names. Which technique would be most effective in safeguarding the application, given the potential for malicious user inputs?

A. Implement a safety filter that detects any harmful inputs and ask the LLM to respond that it is unable to assist

B. Reduce the time that the users can interact with the LLM

C. Ask the LLM to remind the user that the input is malicious but continue the conversation with the user

D. Increase the amount of compute that powers the LLM to process input faster

Suggested Answer: *A*

Community vote distribution

A (100%)

7e28dac Most Recent 1 year, 3 months ago

Selected Answer: A

Safety Filter, i.e. Guard Rails are most effective techniques to detect harmful inputs, and protect the GEN AI application from malicious input.

upvoted 1 times

trendy01 2 years ago

Selected Answer: A

A appears to be the most appropriate choice. Implementing safety filters that detect harmful input can help protect your application from malicious input and ensure safety while maintaining the user experience.

upvoted 2 times

Question #29 *Topic 1*

Which indicator should be considered to evaluate the safety of the LLM outputs when qualitatively assessing LLM responses for a translation use case?

A. The ability to generate responses in code

B. The similarity to the previous language

C. The latency of the response and the length of text generated

D. The accuracy and relevance of the responses

Suggested Answer: *D*

Community vote distribution

D (100%)

trendy01 Most Recent 2 years ago

D appears to be the most appropriate choice. When assessing the safety of an LLM's output, it is important to consider the accuracy and relevance of the responses.

upvoted 1 times

Question #30 *Topic 1*

A Generative AI Engineer is developing a patient-facing healthcare-focused chatbot. If the patient's question is not a medical emergency, the chatbot should solicit more information from the patient to pass to the doctor's office and suggest a few relevant pre-approved medical articles for reading. If the patient's question is urgent, direct the patient to calling their local emergency services.
Given the following user input:
"I have been experiencing severe headaches and dizziness for the past two days."
Which response is most appropriate for the chatbot to generate?

A. Here are a few relevant articles for your browsing. Let me know if you have questions after reading them.

B. Please call your local emergency services.

C. Headaches can be tough. Hope you feel better soon!

D. Please provide your age, recent activities, and any other symptoms you have noticed along with your headaches and dizziness.

Suggested Answer: *B*

Community vote distribution



kirangud77 Most Recent 11 months, 4 weeks ago

Selected Answer: B

It is severe headache --

upvoted 1 times

Vas1l1k0s 1 year, 1 month ago

Selected Answer: B

I think B is the correct answer.

If the LLM considered the symptoms to not reflect an emergency, then it would solicit info AND give out relevant medical articles. We are given two choices on that, one that handles articles and one that handles details about the symptoms. Which means neither A, nor D are correct IMHO.

upvoted 1 times

ellessayy 1 year, 2 months ago

Selected Answer: D

Only plausible explanation i see is if a patient is having headache or dizziness that mean it is not an emergency. Emergency implies that there will be a potential loss of life if not treated in next 30 mins to 1 hour. For example: Breathing difficulties. So the correct answer should be D to collect more information about the patient.

upvoted 1 times

adaine 1 year, 6 months ago

Selected Answer: B

This question seems to rely on our understanding of medical conditions!

I think the answer should be B. Headaches and dizziness that persist for days could be indicative of a serious condition (e.g., stroke, concussion, or other neurological issues). In cases like this, where the symptoms could be urgent, it's critical to guide the user towards immediate medical attention. Directing the patient to emergency services is the safest option, ensuring they receive prompt help.

upvoted 2 times

CoolSmartDude 1 year, 7 months ago

Selected Answer: B

Definitely B, the key here is "[severe] headaches and dizziness for the past two days". They need to call their local emergency services.

upvoted 1 times

feeeeee 1 year, 8 months ago

Selected Answer: B

I voted B but the answer is somehow personal _:D

upvoted 2 times

trendy01 2 years ago

Selected Answer: D

D appears to be the most appropriate choice. It is important to ask the patient for additional information so you can better understand their situation and relay it to the doctor's office if necessary.

upvoted 4 times

Question #31 *Topic 1*

After changing the response generating LLM in a RAG pipeline from GPT-4 to a model with a shorter context length that the company self-hosts, the Generative AI Engineer is getting the following error:

What TWO solutions should the Generative AI Engineer implement without changing the response generating model? (Choose two.)

```
{"error_code": "BAD_REQUEST", "message": "Bad request: rpc error:
code = InvalidArgument desc = prompt token count (4595) cannot
exceed 4096…"}
```

A. Use a smaller embedding model to generate embeddings

B. Reduce the maximum output tokens of the new model

C. Decrease the chunk size of embedded documents

D. Reduce the number of records retrieved from the vector database

E. Retrain the response generating model using ALiBi

Suggested Answer: *C, D*

Community vote distribution

CD (100%)

Lg22 Most Recent 1 year, 6 months ago

C,D should be

upvoted 1 times

trendy01 2 years ago

C: You can reduce the number of tokens included in the prompt by reducing the chunk size of the embedded document. D: Reducing the number of records retrieved from the vector database can solve the token overflow issue by reducing the size of the final prompt.

upvoted 3 times

Question #32 *Topic 1*

A Generative AI Engineer is building a system which will answer questions on latest stock news articles.
Which will NOT help with ensuring the outputs are relevant to financial news?

A. Implement a comprehensive guardrail framework that includes policies for content filters tailored to the finance sector.

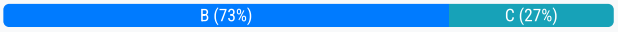
B. Increase the compute to improve processing speed of questions to allow greater relevancy analysis

C. Implement a profanity filter to screen out offensive language.

D. Incorporate manual reviews to correct any problematic outputs prior to sending to the users

Suggested Answer: *C*

Community vote distribution



maciejmirski Highly Voted 2 years ago
I think B will help least with the problem stated
upvoted 6 times

vn327 Most Recent 8 months, 2 weeks ago
B is better answer, more compute just faster, wont make it relverant to finance.. C is filter bad words… finance ppl like bad words…. and we need filter for that.
upvoted 2 times

kirangud77 11 months, 4 weeks ago
will NOT help --C is correct
upvoted 1 times

babi00 1 year, 2 months ago
relevant finance document obviously will not contain any offensive language so profanity filter will actually help getting relevant document , its B that will not help
upvoted 2 times

7e28dac 1 year, 3 months ago
The question clearly mentions - which of them will not help. Profanity Filter will help instead. So i believe the correct answer is B
upvoted 2 times

CoolSmartDude 1 year, 7 months ago
C, it's the only answer that's relevant here
upvoted 1 times

suranga4 1 year, 7 months ago
I think the answer is B because even if we increase the compute it will not effect in getting outputs are relvant to financial news. C is not the answer because implement a Profanity filter to screen out offesnive language will help to get outpts relevant to financial news.
upvoted 3 times

feeeeee 1 year, 8 months ago
I mean profanity filter seems out of scope to me, is NOT of any help
upvoted 1 times

Soumak 1 year, 10 months ago
Compute Power and Relevancy:Increasing compute power can help the model process inputs faster, but it does not inherentlyimprove therelevanceof the answers. Relevancy depends on the data sources, the retrieval method, and the filtering mechanisms in place, not on how quickly the model processes the query.
upvoted 4 times

trendy01 2 years ago
C appears to be the most appropriate answer. Nasty word filters can improve the quality of a system, but they do not guarantee that the output is relevant to financial news.
upvoted 4 times

Question #33 *Topic 1*

A Generative AI 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.

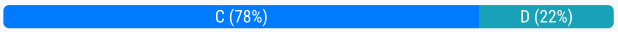
B. Include in the system prompt that any information it sees will be about SnoPenAI, even if no data filtering is performed.

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.

Suggested Answer: *C*

Community vote distribution



nedlo Most Recent 9 months, 3 weeks ago
D is not ideal but still: retrieving happens before any prompt is send to LLM. So if context retrieval retrieves irrelevant context (not containing any info about Snowpen company) and put its into user prompt, then how will system prompt help you in any way?
upvoted 1 times

cooltech 11 months, 2 weeks ago
This prevents LLMs from picking documents not related to snopen in case of some leaks, effective gaurdrail why do people think D. u cannot have docs in 1 chunk. design crash
upvoted 2 times

kirangud77 11 months, 4 weeks ago
Answer - C
upvoted 2 times

e7ec3f6 1 year ago
You should only answer questions about SnoPen AI. Ignore all other topics
upvoted 2 times

gs17 1 year ago
Including a clear instruction in the system prompt helps guide the LLM to focus only on SnoPen AI-related information, effectively filtering out irrelevant content during response generation. This is a common and effective way to reduce hallucination or off-topic answers without needing to preprocess or discard large amounts of data.
upvoted 1 times

adaine 1 year, 6 months ago
D. By keeping all the SnoPen AI content in a single chunk, you're making sure that the retriever pulls in the entire relevant context needed for an accurate response, without mixing in noisy, irrelevant data. (with the main caveat is that if the chunk size becomes too large for the model's context window, you would run into problems).
upvoted 1 times

Question #34 *Topic 1*

A Generative AI Engineer has successfully ingested unstructured documents and chunked them by document sections. They would like to store the chunks in a Vector Search index. The current format of the dataframe has two columns: (i) original document file name (ii) an array of text chunks for each document.
What is the most performant way to store this dataframe?

A. Split the data into train and test set, create a unique identifier for each document, then save to a Delta table

B. Flatten the dataframe to one chunk per row, create a unique identifier for each row, and save to a Delta table

C. First create a unique identifier for each document, then save to a Delta table

D. Store each chunk as an independent JSON file in Unity Catalog Volume. For each JSON file, the key is the document section name and the value is the array of text chunks for that section

Suggested Answer: *B*

Community vote distribution

B (100%)

trendy01 Most Recent 2 years ago

Selected Answer: B

B appears to be the most appropriate choice. By flattening your dataframe into one row per chunk and creating a unique identifier for each row, you can maximize search performance and maintain a manageable structure for your data.

upvoted 1 times

Question #35 *Topic 1*

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.

Suggested Answer: *B*

Community vote distribution



gs17 Most Recent 1 year ago

Selected Answer: B

Separately evaluating retrieval and generation helps pinpoint which part of the RAG pipeline needs improvement.

Curating a test dataset with labeled queries and expected answers allows systematic, repeatable evaluation.

Using tools like MLflow's evaluation metrics supports tracking performance over time and across experiments.

This approach is more thorough and practical than relying solely on similarity scores or subjective assessments.

upvoted 1 times

fa2bede 1 year, 11 months ago

Using LLMs-as-a-judge for our document-based chatbot evaluation is as effective as human judges, according to databricks blogs. option B, references ML flow model evaluation metrics which does not work well for RAG/GenAI model evaluationsI stand corrected, MLflow supports llm as a judge and other evaluation metrics suited for LLM evaluation. Hence Option B is correct

upvoted 1 times

fa2bede 1 year, 11 months ago

Selected Answer: D

Using LLMs-as-a-judge for our document-based chatbot evaluation is as effective as human judges, according to databricks blogs. option B, references ML flow model evaluation metrics which does not work well for RAG/GenAI model evaluations

upvoted 1 times

fa2bede 1 year, 11 months ago

I stand corrected, MLflow supports llm as a judge and other evaluation metrics suited for LLM evaluation. Hence Option B is correct

upvoted 2 times

trendy01 2 years ago

Selected Answer: B

B appears to be the most appropriate choice. By evaluating the search and generation components separately, you can gain a clearer understanding of your system's performance and effectively identify areas for improvement.

upvoted 3 times

Question #36 *Topic 1*

A Generative AI Engineer has already trained an LLM on Databricks and it is now ready to be deployed.
Which of the following steps correctly outlines the easiest process for deploying a model on Databricks?

A. Log the model as a pickle object, upload the object to Unity Catalog Volume, register it to Unity Catalog using MLflow, and start a serving endpoint

B. Log the model using MLflow during training, directly register the model to Unity Catalog using the MLflow API, and start a serving endpoint

C. Save the model along with its dependencies in a local directory, build the Docker image, and run the Docker container

D. Wrap the LLM's prediction function into a Flask application and serve using Gunicorn

Suggested Answer: *B*

Community vote distribution

B (100%)

trendy01 Most Recent 2 years ago

Selected Answer: B

B appears to be the most appropriate choice. Recording and registering your model using MLflow is the most efficient and simple way to deploy your model in Databricks.

upvoted 1 times

Question #37 *Topic 1*

A Generative AI Engineer developed an LLM application using the provisioned throughput Foundation Model API. Now that the application is ready to be deployed, they realize their volume of requests are not sufficiently high enough to create their own provisioned throughput endpoint. They want to choose a strategy that ensures the best cost-effectiveness for their application.
What strategy should the Generative AI Engineer use?

A. Switch to using External Models instead

B. Deploy the model using pay-per-token throughput as it comes with cost guarantees

C. Change to a model with a fewer number of parameters in order to reduce hardware constraint issues

D. Throttle the incoming batch of requests manually to avoid rate limiting issues

Suggested Answer: *B*

Community vote distribution

B (100%)

Maddykrish1984 Most Recent 1 year, 2 months ago

Selected Answer: B

It's the cost effective approach. Also same model can be reused

upvoted 1 times

trendy01 2 years ago

Selected Answer: B

B appears to be the most appropriate choice. A strategy that can maximize cost effectiveness in low request volume situations is to deploy in a usage-based token throughput manner.

upvoted 2 times