Quiz

Essential hands-on tasks quiz [watsonx.ai PoX L4]

Congratulations, you passed!

Your score	Passing score	Date
80% (8 of 10) answered correctly	**75%**	**08 Dec 2023**

Review quiz results

2 incorrect answers

Question 3

With all the hype surrounding large language models (LLMs), it is tempting to assume a LLM-based solution will generally be better than a machine learning (ML)-based solution. This is not always the case: these technologies have different characteristics, making them each suited for different scenarios. A good example is the classification use case. When would it make sense to train a traditional ML-based classifier instead of deploying a LLM-based classifier?

- ◯ When there are only small volumes of training data (where "small" means less than 100 rows).
- ◯ When you need a quick and easy way to develop the solution.
- ✗ ◉ When there are limited computational resorces available for model scoring / inferencing.
- ✓ ◯ When the cost of inferencing/scoring is low and there is a need for fast performance.

People can interact with large language models (LLMs) by issuing instructions and asking questions through a technique called *prompting*. There are multiple prompting approaches. Which one of the following describes "few-shot prompting"?

- ○ Few-shot prompting involves giving LLMs a large dataset to improve their language understanding.
- ✓ ○ Few-shot prompting includes a prompt instruction combined with a few examples of how you want the model to respond.
- ○ Few-shot prompting includes a set of multiple prompts that share the same intention, but each is written in a slightly different way.
- ✗ ◉ Few-shot prompting is a policy that limits the number and rate of prompts that users can issue to a LLM.

✓ 8 correct answers

Question 1

Which one of the models included in watsonx.ai will yield the best results for zero-shot prompting?

- ✓ ◉ llama-2-70b-chat
- ○ mpt-7b-instruct2
- ○ flan-t5-xxl-11b
- ○ starcoder-15.5b

Question 2

Web developers have the ability to infuse prompts developed in watsonx.ai into their applications through a REST interface. What parameters do you need to pass a watsonx.ai model's inferencing endpoint in order for your application to get responses for text that you input to the prompt?

- ○ User id, model id, prompt input, model parameters

- ◯ Model id, prompt id, project id
- ◯ User id, project id, prompt id

Question 4

When preparing a retrieval-augmented generation (RAG) application, one important step is to load data into the knowledge base, which, in many cases, is a vector database. For reasonable performance, data that is loaded into a vector database needs to be divided into individual sections for more efficient retrieval and processing by the LLM. What is this process of dividing the data into sections called?

- ◯ Sharding
- ◯ Partitioning
- ✓ ◉ Chunking
- ◯ Slicing

Question 6

Large language models (LLMs) deployed in watsonx.ai can be invoked either with REST calls or through a Python API. Python code can either be run in scripts or from within Jupyter notebooks. When does it make the most sense to make prompting calls to watsonx.ai LLMs from Python scripts, as opposed to Jupyter notebooks?

- ◯ During the development process for your prompts
- ◯ When you are experimenting with your prompts.
- ✓ ◉ In an application that does real-time inferencing of your prompts.
- ◯ When interactive exploration and step-by-step execution of code is required.

Question 7

Large language models (LLMs) are designed to accept free-form text as an input and generate free-form text as an output. However, the amount of text a LLM can handle can differ depending on the specific model you are using. Therefore, this is an important consideration

- ◯ Query window, number of characters
- ◯ Input window, number of words
- ◯ Parameter buffer, number of bytes
- ✓ ◉ Context window, number of tokens

Question 8

Large language models (LLMs) are *stateless*, which means that previous prompts have no bearing on future prompts. In other words, if you are to issue a series of prompts with a LLM, there would be no accumulation of context while the conversation progresses. You are building a LLM-based application that has a chatbot-like experience, where maintaining context as the coversation continues is essential. What technique would you use to enable your LLM-based application to preserve context?

- ◯ Ask the user to preface each new prompt with a summary of the most recent exchanges.
- ✓ ◉ Use the Conversation Buffer capability in LangChain to record conversation context.
- ◯ Set the "statefulness" parameter to "true" for all the calls your application makes to the LLM.
- ◯ Use a LLM that has built-in memory for storing conversation context.

Question 9

The watsonx.ai Prompt Lab has a model parameters dialog where you can tune the behavior of their selected large language model (LLM). You are working on a project where you need the text generated for your application to use a wider vocabulary. What setting do you apply in the Prompt Lab to ensure your answers have greater variability of the words included in the generated text?

- ◯ Set the "Decoding" setting to "Verbose".
- ✓ ◉ Set the "Decoding" setting to "Sampling".
- ◯ Set the "Encoding" setting to "Variety".

Question 10

For retrieval-augmented generation (RAG) applications, challenges often arise when the information you need to load into the knowledge base is stored in binary document formats, like PDF. There are frequently errors when extracting text from binary documents, especially if the contents of the files are more complex. With this in mind, which one of the following approaches is best for ingesting data stored in document formats like PDF, Excel, and Word?

- ✓ (●) Extract text using Watson Discovery.
- ( ) Generate tokens from the documents using watsonx.ai.
- ( ) Apply the embedding model directly against the documents.
- ( ) Store the documents in binary large object (BLOB) columns in a database.

Done