Lab: Get Familiar with GenAI Classroom

Skills
Network

Estimated Time: 20 mins

Introduction:

Within the Generative AI Classroom, you have the opportunity to delve into the art of prompt engineering. Here, you can actively engage in writing and comparing your prompts, allowing you to refine your skills through real-time chat responses.Additionally, you'll have the option to utilize the GPT-4o model, an upgraded version of OpenAI's language model. This enhanced iteration is specifically designed to significantly enhance its ability in generating text that closely resembles human language while comprehending and adapting to various tasks and situations.

Learning Objectives:

After completing this lab, you should be able to perform the following tasks:

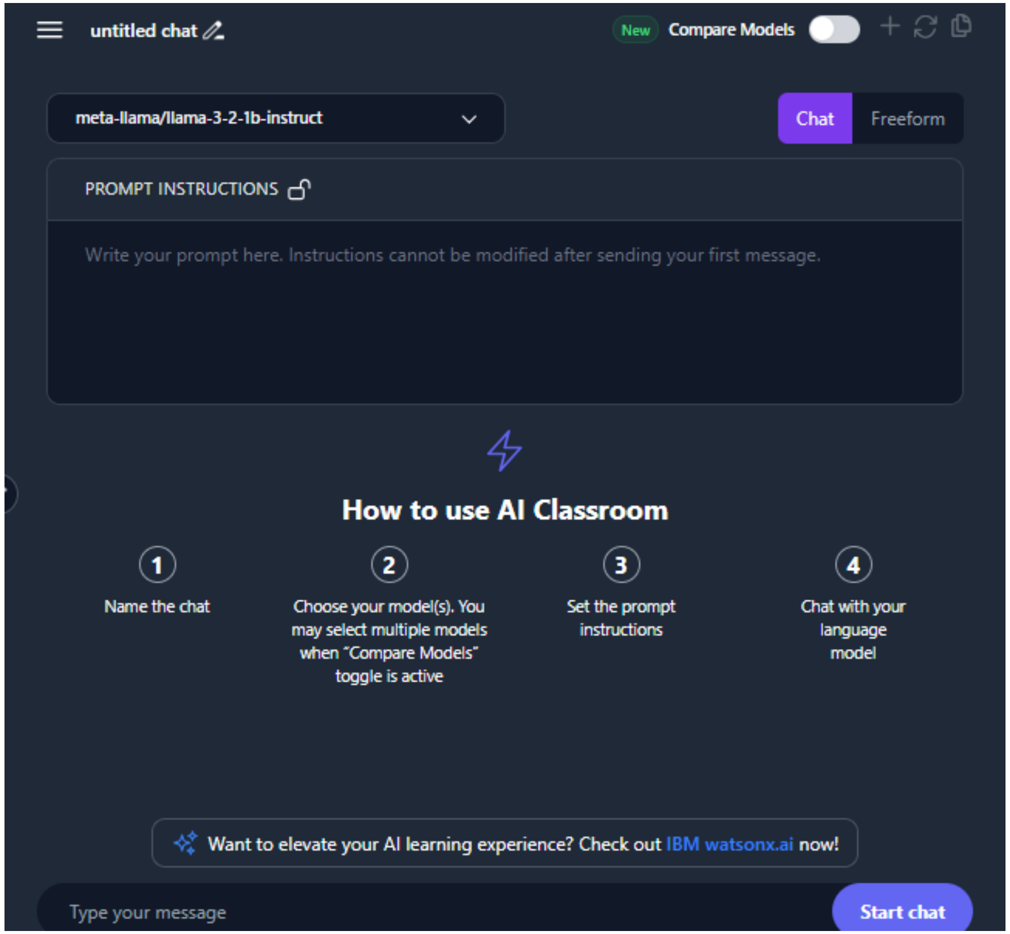
- Develop a working understanding of the GenAI Classroom platform, exploring its features and tools tailored for learning prompt engineering.
- Engage in a conversation with the chatbot, asking various questions to observe how it responds accurately, showcasing the functionality of these prompts.

  Please note that generative AI is an evolving field. As you attempt the labs, your experience and output might be different than what is seen here.

Task 1: Set up your AI Classroom

Within a lab environment, you will have a single-page display featuring both the Generative AI Classroom interface and your instructional content. The Instructions will be positioned on the left-hand side, while the Generative AI Classroom environment will be accessible on the right-hand side of the same page. This layout facilitates simultaneous access to guidance and the interactive learning environment, enhancing convenience and ease of use.

1. The desired appearance of your AI Classroom should reflect this.



Write your prompt here. Instructions cannot be modified after sending your first message.

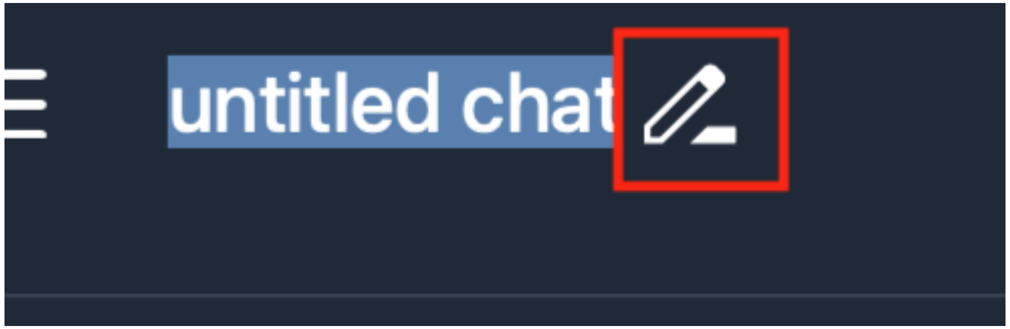
How to use AI Classroom

1. Name the chat
2. Choose your model(s). You may select multiple models when "Compare Models" toggle is active
3. Set the prompt instructions
4. Chat with your language model

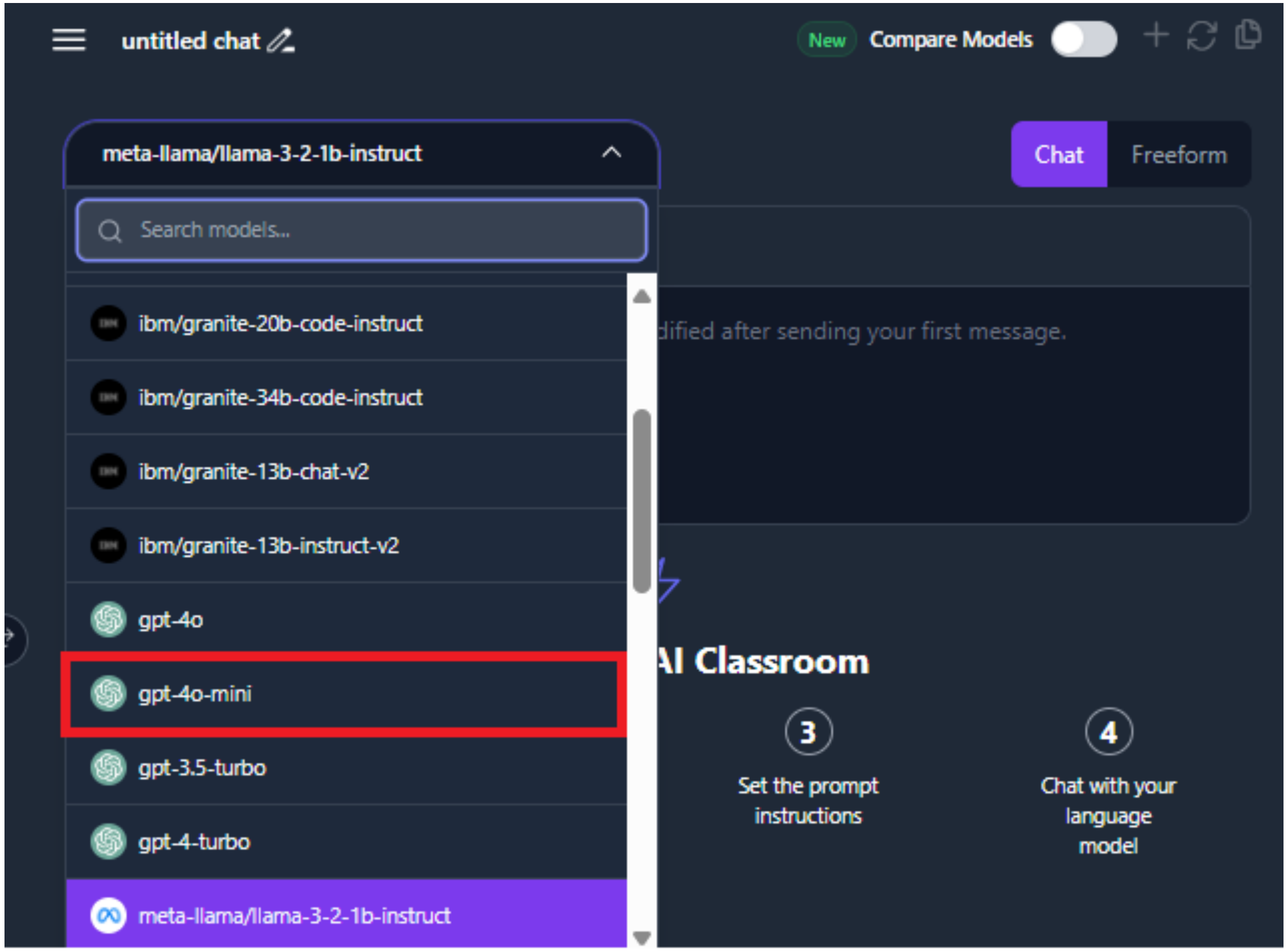
Want to elevate your AI learning experience? Check out IBM watsonx.ai now!

Type your message

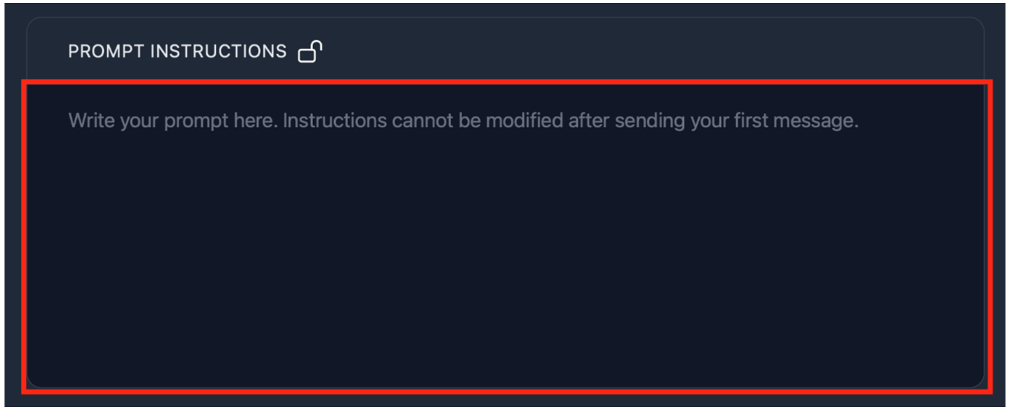
2. **Name the chat:** Utilize the pencil icon found at the top-left corner of the right pane to assign a name to the chat.



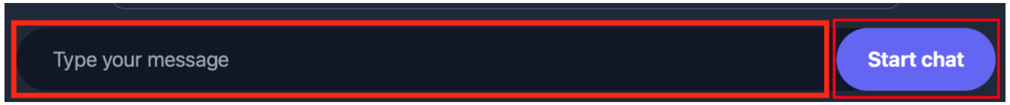
3. **Select the model:** Utilize the dropdown menu at the top-right corner of the pane to choose the GPT-4o mini AI model for chat interactions. Here, by default, the model **meta-llama-3-2-1b-instruct** is selected.


untitled chat
New
Compare Models
meta-llama/llama-3-2-1b-instruct
Chat
Freeform
Search models...
ibm/granite-20b-code-instruct
ibm/granite-34b-code-instruct
ibm/granite-13b-chat-v2
ibm/granite-13b-instruct-v2
gpt-4o
gpt-4o-mini
gpt-3.5-turbo
gpt-4-turbo
meta-llama/llama-3-2-1b-instruct
dified after sending your first message.
AI Classroom
3
Set the prompt instructions
4
Chat with your language model

4. **Provide prompt guidance**: Access the Prompt Instructions field located in the upper right pane of the chat system to input specific instructions or contextual details regarding the required queries. Once the chat begins, the prompt will be fixed and cannot be altered thereafter.


PROMPT INSTRUCTIONS
Write your prompt here. Instructions cannot be modified after sending your first message.

5. **Enter your message:** Utilize the text box situated at the bottom of the page to input your question and interact with the chat system.



Once you have entered the **Type your message** field, proceed by clicking the 'Start Chat' button to initiate the conversation.

Task 2: Posing queries to ChatGPT

Once you have configured the AI Classroom, let's move forward and pose some questions to ChatGPT.

1. Please enter your query in the **Type your message** field to request guidance from ChatGPT regarding palindromes.

Question to chatgpt:

“What is a palindrome and how is it defined in terms of string or number sequences?”

▾ Click here to view the sample response generated

A palindrome is a sequence of characters, whether in the form of a string or a number, that reads the same forward and backward. In other words, if you reverse the sequence, it remains unchanged.

For example:

- In strings: “radar”, “level”, and “madam” are all palindromes.
- In numbers: 121, 1331, and 12321 are examples of palindromic numbers.

Additional Details:

1. **Character Types**: Palindromes can include letters, numbers, and even punctuation. For instance, “A man, a plan, a canal, Panama!” is a famous palindromic phrase when ignoring spaces and punctuation.

2. **Length**: A palindrome can be of any length, including single-character sequences, which are trivially palindromic.

3. **Applications**: Palindromes are often used in computer science for algorithms related to string processing, data validation, and cryptography. They also appear in recreational mathematics and puzzles.

4. **Checking for Palindromes**: In programming, you can check if a string is a palindrome by comparing it to its reverse. For example, in Python:

   ```
   def is_palindrome(s):
       return s == s[::-1]
   ```

Understanding palindromes can also enhance your skills in pattern recognition and algorithmic thinking, making them a fun topic to explore!

2. Please enter your query in the **Type your message** field to seek guidance from ChatGPT about the contrast between JavaScript and Java.

Question to chatgpt:

“What is the primary dissimilarity between JavaScript and Java?”

▾ Click here to view the sample response generated

The primary dissimilarity between JavaScript and Java lies in their design, usage, and execution environments.

Key Differences:

1. **Type of Language**:
    - **Java**: A statically typed, object-oriented programming language. It requires explicit declaration of variable types, and it enforces a strict structure.
    - **JavaScript**: A dynamically typed, prototype-based scripting language. Variable types can change at runtime, and it is more flexible in terms of structure.

2. **Execution Environment**:
    - **Java**: Typically runs on the Java Virtual Machine (JVM), which allows Java applications to be platform-independent. Java is often used for server-side applications, Android development, and large-scale enterprise systems.
    - **JavaScript**: Primarily runs in web browsers, allowing for dynamic content on web pages. It is mainly used for client-side scripting, although with environments like Node.js, it can also be used for server-side development.

3. **Syntax and Structure**:
    - **Java**: Has a syntax similar to C++, with a focus on classes and objects. It uses a more verbose syntax and requires the definition of classes and methods.
    - **JavaScript**: Uses a more flexible syntax that allows for functional programming styles and does not require class definitions for objects. It supports first-class functions and closures.

4. **Concurrency Model**:
    - **Java**: Uses multithreading and synchronization for concurrent execution, which can be complex to manage.

    - **JavaScript**: Uses an event-driven, non-blocking I/O model with a single-threaded event loop, which simplifies handling asynchronous operations.

Additional Details:

- **Use Cases**: Java is commonly used in large-scale enterprise applications, mobile applications (especially Android), and backend systems. JavaScript is essential for web development, enabling interactive and dynamic features on websites, and is increasingly used in full-stack development with frameworks like React, Angular, and Vue.js.

- **Community and Ecosystem**: Both languages have large communities and extensive ecosystems. Java has a rich set of libraries and frameworks like Spring and Hibernate, whereas JavaScript has a vast array of libraries and frameworks, including jQuery, Express.js, and Node.js.

- **Learning Curve**: Java’s strict typing and structured approach can make it easier for beginners to grasp programming concepts, while JavaScript’s flexibility allows for quick prototyping and experimentation, but may lead to more complex debugging.

Understanding these differences can help you choose the right language for your specific project needs and goals!

3. Please enter your query in the **Type your message** field to request advice from ChatGPT regarding software development methodologies.

Question to chatgpt:

“What are the primary software development methodologies and their key differences?”

▾ Click here to view the sample response generated

The primary software development methodologies include Agile, Waterfall, DevOps, Scrum, Kanban, and Lean. Each methodology has its own principles, practices, and approaches to project management and software development.

Key Methodologies:

1. **Agile**:

   - **Overview**: Agile is an iterative and incremental approach that emphasizes flexibility, collaboration, and customer feedback. It promotes adaptive planning and encourages rapid delivery of functional software.
   - **Key Features**: Short development cycles called sprints, regular feedback loops, and continuous improvement.
   - **Use Cases**: Suitable for projects where requirements may change frequently.

2. **Waterfall**:

   - **Overview**: Waterfall is a linear and sequential approach where each phase of development must be completed before the next begins. It follows a structured path from requirements gathering to design, implementation, testing, and maintenance.
   - **Key Features**: Clear documentation, defined stages, and a focus on upfront planning.
   - **Use Cases**: Best for projects with well-defined requirements and low likelihood of changes.

3. **DevOps**:

   - **Overview**: DevOps is a cultural and technical movement that aims to improve collaboration between development and operations teams. It focuses on automating processes and integrating continuous development, testing, and deployment.
   - **Key Features**: Continuous integration and continuous deployment (CI/CD), infrastructure as code, and monitoring.
   - **Use Cases**: Ideal for organizations looking to improve software delivery speed and reliability.

4. **Scrum**:

   - **Overview**: Scrum is a framework within Agile that organizes work into time-boxed iterations called sprints, typically lasting two to four weeks. It includes roles such as Scrum Master and Product Owner.
   - **Key Features**: Daily stand-up meetings, sprint reviews, and retrospectives to foster team collaboration and accountability.
   - **Use Cases**: Effective for teams needing to manage complex projects with changing requirements.

5. **Kanban**:

   - **Overview**: Kanban is a visual workflow management method that focuses on continuous delivery without overloading team members. It uses a Kanban board to visualize tasks and their progress.
   - **Key Features**: Limiting work in progress (WIP), visualizing workflow, and managing flow.
   - **Use Cases**: Useful for teams that need to manage ongoing work and prioritize tasks dynamically.

6. **Lean**:
   - **Overview**: Lean development focuses on maximizing customer value while minimizing waste. It aims to optimize efficiency and improve the flow of work.
   - **Key Features**: Emphasizes continuous improvement, reducing unnecessary steps, and delivering value quickly.
   - **Use Cases**: Suitable for organizations looking to streamline processes and enhance productivity.

Additional Details:

- **Flexibility vs. Structure**: Agile, Scrum, and Kanban are more flexible and adaptable to changes, while Waterfall provides a structured approach that requires thorough planning upfront.

- **Team Collaboration**: Agile and Scrum emphasize team collaboration and stakeholder involvement throughout the development process, whereas Waterfall may have less frequent interaction with stakeholders after the initial requirements phase.

- **Speed of Delivery**: DevOps and Agile methodologies focus on delivering software quickly and iteratively, while Waterfall may have longer delivery cycles due to its sequential nature.

- **Choosing a Methodology**: The choice of methodology often depends on the project requirements, team structure, and organizational culture. Many organizations adopt hybrid approaches, combining elements from different methodologies to suit their needs.

Understanding these methodologies can help teams select the most appropriate approach for their projects, leading to improved efficiency, collaboration, and successful outcomes.

Summary

Throughout this lab, you've delved into the potential of generative AI models for text generation. You've familiarized yourself with IBM's Generative AI classroom chatbot system, driven by multiple language models. You've also gained knowledge on creating desired text outputs using prompts and explored the text generation capabilities inherent in language models.

Congratulations!

You've completed the practical session aimed at acquainting yourself with the GenAi classroom setup and practicing text generation through generative AI.

Author(s)

Pallavi Rai