SAP AI Launchpad shared-practice-ai-core (Learning-AIG) AS

Áreas de trabalho

Hub de IA generativa

- Biblioteca de modelos ●
- Gerenciamento de embasame...
- Chat
- Editor de prompts
- Orquestração
- Administração de prompts
- Otimização

IA generativa / Biblioteca de modelos

Biblioteca de modelos

Explore os modelos disponíveis e suas especificações. Infor...

F.

Modelos (44) Modo: **Catálogo**

Selecione um cartão de modelo para visualizar os detalhes.

Visualizar configurações:
Tudo ✕ Hide Deprecated ✕

Claude 3 Haiku
(anthropic--claude-3-haiku)
Versão: 1

Claude 3.7 Sonnet
(anthropic--claude-3.7-sonnet)
Versão: 1

Claude 4 Sonnet
(anthropic--claude-4-sonnet)
Versão: 1

Claude 4.5 Haiku
(anthropic--claude-4.5-haiku)
Versão: 1

SAP AI Launchpad shared-practice-ai-core (Learning-AIG) AS

Áreas de trabalho

Conexões API AI (1)

Selecione uma conexão API AI como sua área de trabalho.

shared-practice-ai-core

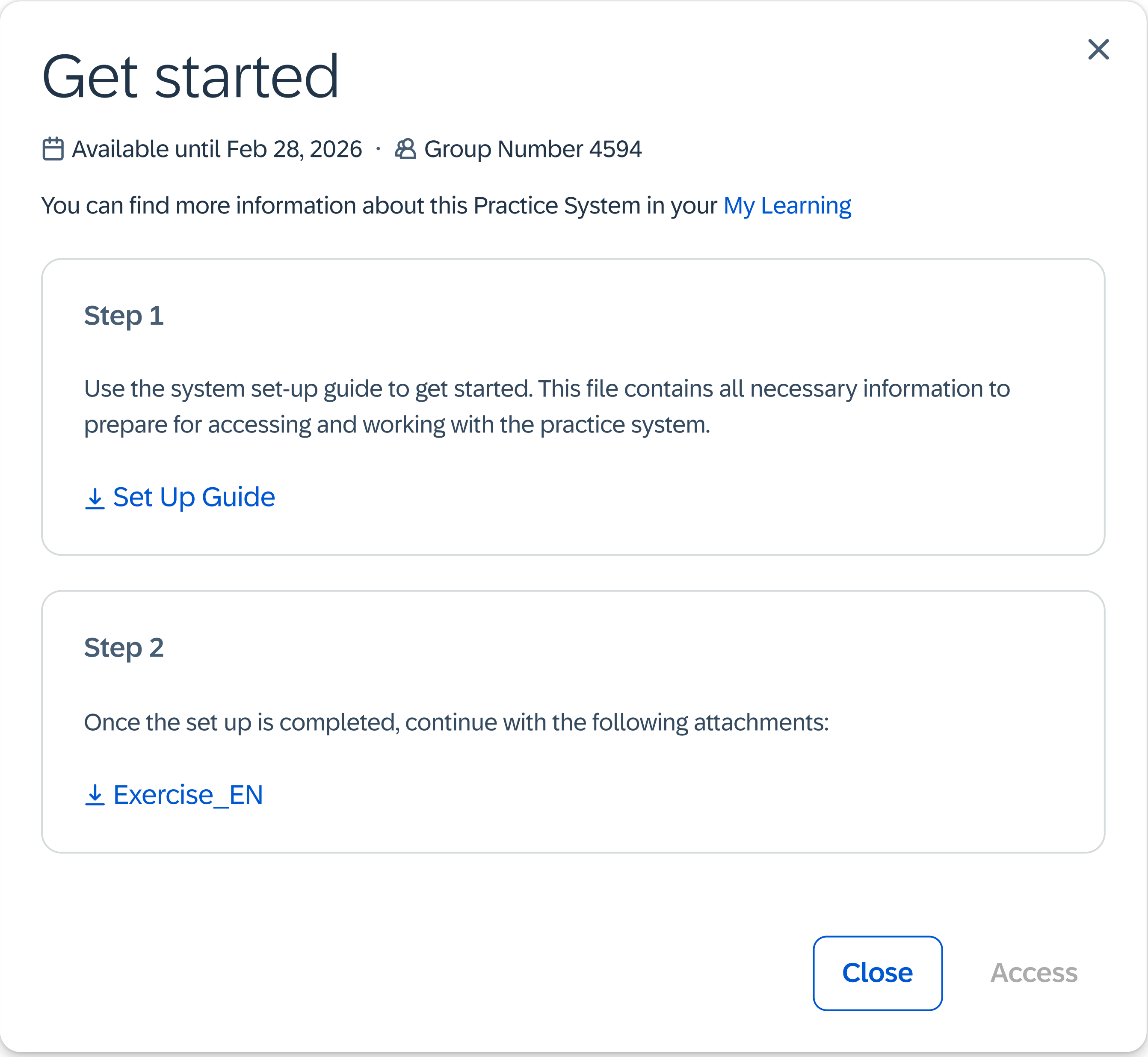
ID do inquilino
0f53ab30-9ab8-458f-84d1-fd89cd390595

Grupc

Selecione um grupo de recursos como sua área de trabalho.

Learning-AIG

Criado em:
8 de out. de 2025



Get started

Available until Feb 28, 2026 · Group Number 4594

You can find more information about this Practice System in your My Learning

Step 1

Use the system set-up guide to get started. This file contains all necessary information to prepare for accessing and working with the practice system.

Set Up Guide

Step 2

Once the set up is completed, continue with the following attachments:

Exercise_EN

Get started

Available until Feb 28, 2026 · Group Number 4594

You can find more information about this Practice System in your My Learning

Step 1

Use the system set-up guide to get started. This file contains all necessary information to prepare for accessing and working with the practice system.

⤓ Set Up Guide

Step 2

Once the set up is completed, continue with the following attachments:

⤓ Exercise_EN

Close Access

Get started

Available until Feb 28, 2026 · Group Number 4594

You can find more information about this Practice System in your My Learning

Step 1

Use the system set-up guide to get started. This file contains all necessary information to prepare for accessing and working with the practice system.

Set Up Guide

Step 2

Once the set up is completed, continue with the following attachments:

Exercise_EN

Close Access

Learning Browse Get Certified My Learning Subscribe Explore SAP

/ Browse / Courses / Using Advanced AI Techniques with SAP's Generative AI Hub...

Knowledge quiz

It's time to put what you've learned to the test, get 3 right to pass this unit.

1. Which sequence of steps are used in the Document Grounding module as part of the orchestration service to generate content with the RAG approach?

Choose the correct answer.

○ Configure the Document Grounding module, create the knowledge base, and finally generate content using the RAG approach based on the knowledge base.

○ Generate content using the RAG approach based on the knowledge base and Configure the Document Grounding module. Note that the knowledge base is automatically created.

● Create the knowledge base, then Configure the Document Grounding module and finally generate content using the RAG approach based on the knowledge base.

using the RAG approach based on the knowledge base.

2. Which of the following are options for creating vector embeddings for the Grounding module?

There are two correct answers.

☑ Upload Documents to Supported Data Repository and Run Data Pipeline.

□ Ensure that all documents are of the same type, before copying into a data repository.

☑ Provide Chunks of Documents via Vector API Directly.

Correct

The Grounding module options are Upload Documents to Supported Data Repository and Run Data Pipeline and Provide Chunks of Documents via Vector API Directly.

3. Which function within the SAP HANA Vector Engine is used to calculate the Euclidean distance between vectors?

Choose the correct answer.

○ SQL_QUERY()

○ EMBEDDING_SEARCH()

● L2DISTANCE()

Correct

Correct ! L2DISTANCE() is the function used to calculate the Euclidean distance between vectors, valuable for tasks like clustering and nearest neighbor searches.

4. How does document grounding within the generative AI hub improve AI responses?

Choose the correct answer.

○ By fine-tuning LLMs on proprietary company data.

● By merging LLMs with advanced information retrieval techniques for more accurate responses.

○ By using embedding models to produce text directly.

○ By integrating generative models to create a context-aware environment.

techniques, enhancing the accuracy without extensive fine-tuning.

YOUR SCORE

4/4

You passed Mastering Document Grounding Using Generative AI Hub

2 Lessons 40min

Next up: Analyzing Document Grounding in Generative AI Hub

Try Again

Continue

Learning

Subscribe

/ Browse / Courses / Using Advanced AI Techniques with SAP's Generative AI Hub...

Learning Support

Get Support

Share Feedback

Release Notes

About SAP

Company Information

Copyright

Trademark

Worldwide Directory

Careers

News and Press

Site Information

Privacy

Terms of Use

Legal Disclosure

ot Share/Sell My Personal Information (US Learners Only)

rências de Cookies

Implementing Document Grounding in the Orchestration Service

Objective

After completing this lesson, you will be able to demonstrate the practical application of Document Grounding within the generative AI hub.

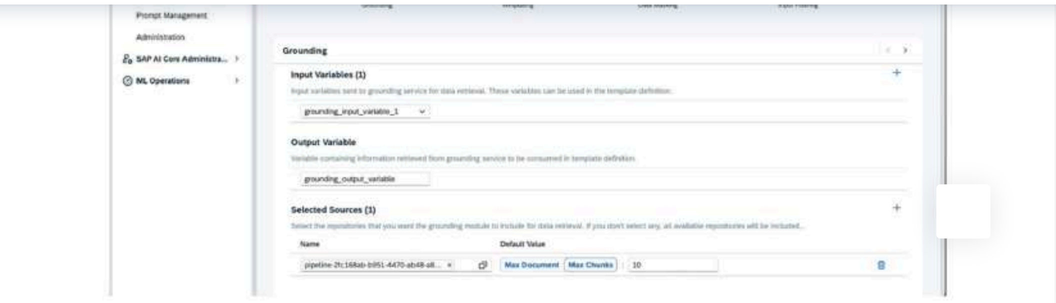
Implementing Document Grounding in the Orchestration Service

You explored the foundational concepts of document grounding, vector embeddings, and the SAP HANA vector engine.

In this lesson, you will discover how to leverage the generative AI hub's **Orchestration Service** to implement document grounding. Using the RAG approach, you will see a step-by-step process of setting up your knowledge base and configuring the grounding module to generate highly accurate and contextually relevant AI responses.

Implementing Document Grounding in the Generative AI Hub

The **grounding capability** is integral to the **orchestration module** of the generative AI hub. This module facilitates specialized data retrieval from vector databases, ensuring the AI's responses are grounded in external, context-relevant information. In addition, the **Pipeline API** integrates vector stores like the managed SAP HANA database, which is directly accessible and works within the SAP Generative AI Hub via SAP AI Core.



Grounding

Input Variables (1)

Input variables sent to grounding service for data retrieval. These variables can be used in the template definition.

grounding_input_variable_1

Output Variable

Variable containing information retrieved from grounding service to be consumed in template definition.

grounding_output_variable

Selected Sources (1)

Select the repositories that you want the grounding module to include for data retrieval. If you don't select any, all available repositories will be included.

Name	Default Value
pipeline-2fc168ab-b951-4470-ab48-a8...	Max Document Max Chunks 10

SAP AI Launchpad also provides the grounding management app lets you manage the lifecycle of your data pipelines.

The generative AI hub supports robust document grounding through several key features and streamlined processes:

- **Access to Diverse LLMs:** The generative AI hub provides instant access to a wide range of LLMs from various providers, including Azure OpenAI models (gpt-5), Anthropic models(anthropic--claude-4-sonnet), and open-source models like Mistral and Meta models. This broad access enables you to orchestrate multiple LLMs to best suit grounding and content generation needs. To see all the available models, refer to 3437766 - Availability of Generative AI Models .
- **Seamless Integration with SAP AI Launchpad:** You can execute and monitor grounded prompts directly within the SAP AI Launchpad. This integration shows how generative AI, combined with your business data, can directly assist business processes while leveraging the underlying SAP AI Core infrastructure for secure operations.
- **Efficient Document Indexing:** Unstructured and semi-structured documents are preprocessed, divided into chunks, and converted to numerical embeddings with embedding models. These embeddings are efficiently stored in the SAP HANA Vector Engine for rapid and precise querying, fundamental to grounding AI responses in real, relevant data.

These integrated features empower you to build generative AI solutions that leverage your organization's trusted document repositories and provide reliable, transparent, and contextually accurate responses.

Grounding for Content Generation

Using the RAG approach, you will utilize the Document Grounding module within the Orchestration Service to generate content effectively. This module aligns user queries with relevant documents by retrieving them

- Create the knowledge base with relevant documents.
- Configure the Document Grounding module in the Orchestration Service.
- Generate content using the RAG approach based on the knowledge base.

Prerequisites

- Install the SAP Cloud SDK for AI (Python) - generative using the command:

Python

```
%pip install "sap-ai-sdk-gen[all]"
```

- Set the credentials for the SDK.

Detailed Steps

Step 1: Create a Vector Knowledge Base

- Prepare your knowledge base before using the Grounding module in the orchestration pipeline.
- The generative AI hub provides several options for users to prepare their knowledge base data:
  - Upload documents to a supported data repository, then run the data pipeline to vectorize the documents. For more details, refer to the Pipeline API .
  - Use the Vector API to directly provide chunks of the document. For additional information, see the Vector API .

Grounding Module Options

Choose one of the following options to use grounding:

Option 1: Upload Documents to Supported Data Repository and Run Data Pipeline

- The pipeline collects documents and segments the data into chunks.

group and a generic secret for grounding. For more information, see:

- Create a Resource Group for Grounding
- Grounding Generic Secrets for Microsoft SharePoint

2. **Prepare Vector Knowledge Base:** Configure the Pipeline API to read unstructured data from data repositories and store it in a vector database. Use the Pipeline API to:
   - Read unstructured documents from various data repositories. Break the data into chunks and create embeddings.
   - Store the multidimensional representations of the textual information in the vector database.
   - Provide a repository ID to access the data.
     For more information, see Preparing Data Using the Pipeline API .

Option 2: Provide Chunks of Documents via Vector API Directly

Provide chunks of data directly and store them using the Vector API. The process involves the following steps:

1. **Perform Initial One-Time Administrative Steps:**Create a Resource Group for Grounding .
2. **Prepare Vector Knowledge Base:** Provide chunks of information directly and store data in the vector database using the Vector API. Use the Vector API to:
   - Create collections.
   - Create documents by directly using the chunks of data provided by users.
   - Store data in the vector database.
   - Assign repository IDs to access the data.
   - For more information, see Preparing Data Using the Vector API .
3. **Configure Grounding Module in the Orchestration:** In the orchestration pipeline, you add configuration for the grounding requests:
   - Create a grounding request configuration in the orchestration pipeline using the repository IDs and filters.
   - Run the orchestration pipeline and check that the response refers to the user data. For more information, see Using the Grounding Module .

Module

Now, you must define the configuration for the Document Grounding module, including setting up filters, and specifying the data repositories.

Python

```python
orchestration_service_url = <your url code from
```

You must have at least one orchestration-compatible deployment for a generative AI model running. For more information, see and Create a Deployment for Orchestration in SAP AI Core .

Next, you must import all relevant libraries. See the code in the code repository here.

Python

```python
# Set up the Orchestration Service
aicore_client = get_proxy_client().ai_core_
orchestration_service = OrchestrationServic
llm = LLM(
    name="gpt-4o",
    parameters={
        'temperature': 0.0,
    }
)
template = Template(
            messages=[
                SystemMessage("""Facility S
                individual homes, and comme
                Customers are encouraged to
                """),
                UserMessage("""You are a he
                Answer the request by provi
                Request: {{ ?user_query }}
```

This Python code sets up an orchestration service crucial for handling complex tasks. It initializes an AI core client and configures an orchestration service with a given URL. The code then sets up an LLM with specific parameters to ensure consistent responses. Lastly, it creates a message template to aid in answering customer inquiries efficiently and effectively.

Next, we set up grounding services.

Python

```
# Set up Document Grounding
filters = [DocumentGroundingFilter(id="vector"
                                   data_reposi
                                   search_conf
                                   data_reposi
                                   )
]

grounding_config = GroundingModule(
            type="document_grounding_service",
            config=DocumentGrounding(input_par
        )

config = OrchestrationConfig(
    template=template,
    llm=llm,
    grounding=grounding_config
)

```

document retrieval based on user inputs.

Step 3: Generate Context-Relevant Answers

Run the orchestration service with the configured settings to generate answers based on user queries.

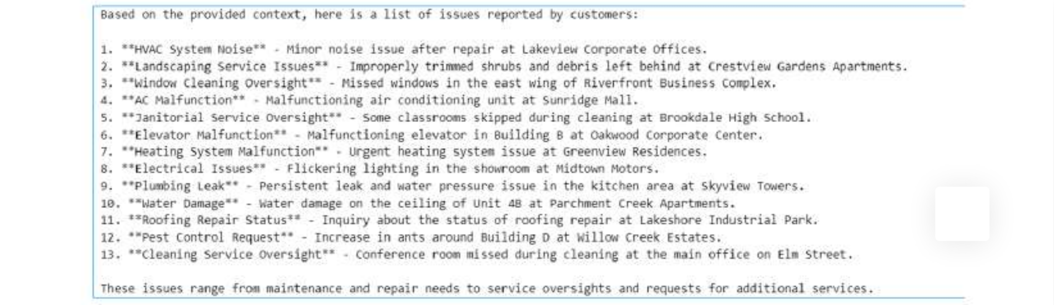
Python

```python
response = orchestration_service.run(config=con
                                     template_values=[
                                         TemplateValue("
                                     ])
print(response.orchestration_result.choices[0].

```

This Python code sends a request to an orchestration service to run a specific configuration. It includes a template value with a user query asking to list customer-reported issues. After running the service, it prints the response from the orchestration result, specifically the message content of the first choice. This helps automate and fetch data on customer issues efficiently.

You get the following output:



```
Based on the provided context, here is a list of issues reported by customers:

1. **HVAC System Noise** - Minor noise issue after repair at Lakeview Corporate Offices.
2. **Landscaping Service Issues** - Improperly trimmed shrubs and debris left behind at Crestview Gardens Apartments.
3. **Window Cleaning Oversight** - Missed windows in the east wing of Riverfront Business Complex.
4. **AC Malfunction** - Malfunctioning air conditioning unit at Sunridge Mall.
5. **Janitorial Service Oversight** - Some classrooms skipped during cleaning at Brookdale High School.
6. **Elevator Malfunction** - Malfunctioning elevator in Building B at Oakwood Corporate Center.
7. **Heating System Malfunction** - Urgent heating system issue at Greenview Residences.
8. **Electrical Issues** - Flickering lighting in the showroom at Midtown Motors.
9. **Plumbing Leak** - Persistent leak and water pressure issue in the kitchen area at Skyview Towers.
10. **Water Damage** - Water damage on the ceiling of Unit 4B at Parchment Creek Apartments.
11. **Roofing Repair Status** - Inquiry about the status of roofing repair at Lakeshore Industrial Park.
12. **Pest Control Request** - Increase in ants around Building D at Willow Creek Estates.
13. **Cleaning Service Oversight** - Conference room missed during cleaning at the main office on Elm Street.

These issues range from maintenance and repair needs to service oversights and requests for additional services.
```

You can see that the list of issues reported by customer, which is grounded in mails that customers provided.

```
print(response.module_results.grounding.data['g
```

This code extracts and displays the value of 'grounding_result' from a nested data structure within the 'response' object. This specific piece of data could be critical for understanding the outcome of a grounding module, making the code essential for debugging or analysis.

The output lists all the relevant mails used for response earlier, providing a deep insight into the context of the grounding technique. You can see this output in the repository.

Conclusion

In this lesson, you have accomplished the following key tasks:

- **Established a Vector Knowledge Base:** You learned to upload and vectorize documents, forming a robust, searchable knowledge base.
- **Configured the Document Grounding Module:** You set up the module within the Orchestration Service to intelligently retrieve relevant documents based on user queries.
- **Generated Grounded AI Responses:** You utilized the Orchestration Service to produce accurate and contextually relevant answers, demonstrating the power of grounding.

Lesson Summary

Building on your understanding of document grounding's foundational concepts, vector embeddings, the SAP HANA vector engine, and RAG, this lesson provided a practical deep dive into implementation. You successfully learned to establish a vector knowledge base, configure the Document Grounding module within the Generative AI Hub's Orchestration Service, and generate accurate, context-relevant AI responses. This end-to-end experience solidified how grounding significantly enhances AI's reliability and relevance by integrating precise, external data into your solutions.

Throughout this learning journey, you started with discovering the significance of SAP Business AI and how it addresses business challenges by automating processes, enhancing decision-making, and supporting scalable, responsible innovation.

You then explored the foundations of LLMs, how they function, their remarkable strengths, and their inherent limitations. You saw how LLMs can automate tasks, generate content, and synthesize information at scale, but also why it's essential to use them thoughtfully, with an awareness of risks like hallucinations, bias, and data privacy. You discovered that practical prompt engineering is the key to guiding LLMs to deliver relevant and reliable results.

You then delved into SAP's generative AI hub, part of AI Foundation, understanding how it is a secure gateway to enterprise-grade AI. The hub empowers you to access and orchestrate leading LLMs, manage prompts, and confidently integrate AI into business processes. It ensures that your solutions are grounded in a real business context, reliable through robust security and compliance, and responsible by design.

Building on this foundation, you learned practical techniques for solving real business problems with LLMs, from prompt development and refinement to integrating prompts into applications using SDKs. You understood the art of prompt engineering, version control, and evaluation, ensuring your AI solutions are accurate, actionable, and scalable. The scalability is supported by creating and managing prompt templates in generative AI hub. Advanced techniques like few-shot prompting and meta prompting further enhanced the quality and consistency of AI outputs, enabling you to tackle even more complex scenarios.

Finally, you discovered how document grounding connects LLMs to your organization's trusted knowledge sources, ensuring accurate and context-aware responses. You can transform documents into searchable vectors by leveraging SAP HANA Vector Engine and embedding models, enabling semantic search and RAG. Grounding minimizes hallucinations and aligns AI outputs with real business facts, making your solutions more trustworthy and effective.

In summary, you are now equipped to design, build, and deploy intelligent, reliable, and business-ready AI solutions, combining the power of LLMs,

Learning Subscribe

/ Browse / Courses / Using Advanced AI Techniques with SAP's Generative AI Hub...

SAP Learning

Quick links

Download Catalog (CSV, JSON, XLSX, XML)

SAP Learning Hub

SAP Training Shop

SAP Developer Center

SAP Community

Newsletter

Learning Support

Get Support

Share Feedback

Release Notes

About SAP

Company Information

Copyright

Trademark

Worldwide Directory

Careers

News and Press

Site Information

Privacy

Terms of Use

Legal Disclosure

Do Not Share/Sell My Personal Information (US Learners Only)

Preferências de Cookies

Learning Subscribe

/ Browse / Courses / Using Advanced AI Techniques with SAP's Generative AI Hub...

Analyzing Document Grounding in Generative AI Hub

Objective

After completing this lesson, you will be able to analyze SAP HANA vector engine and document grounding.

Analyzing Document Grounding in Generative AI Hub

Integrating proprietary data for accurate and context-aware responses is important to effectively leverage generative AI in enterprise solutions. This unit focuses on Document Grounding with SAP's generative AI hub, a key technique for achieving high precision and reliability when combining Large Language Models (LLMs) with business data.

In this lesson, you will learn about **SAP HANA vector engine** and the core principles behind **document grounding.** You will begin by continuing the practical facility management scenario to illustrate real-world business challenges that grounding effectively addresses. This will set the stage for implementing these powerful techniques in the orchestration service, which we will delve into in our next lesson.

Scenario: Facility Management Optimization

Facility Solutions Company delivers comprehensive facility management, maintenance, and cleaning services for residential and commercial properties. Their mission is to create safe, efficient, and impeccably maintained environments, allowing clients to focus on core activities. The

The company receives thousands of emails daily, encompassing customer requests, complaints, and general inquiries. Manually processing these emails, which involves transferring data to internal applications, categorizing, and prioritizing tasks, is time-consuming, prone to errors, and frequently causes delays in addressing critical customer needs.

Previous AI Enhancements:

The company previously leveraged generative AI and prompt engineering to improve information extraction from customer emails. This allowed them to query AI models for specific tasks, leading to more accurate responses and enhanced efficiency in categorizing and prioritizing customer requests and complaints.

Evolving Challenges and Generative AI Hub Solutions:

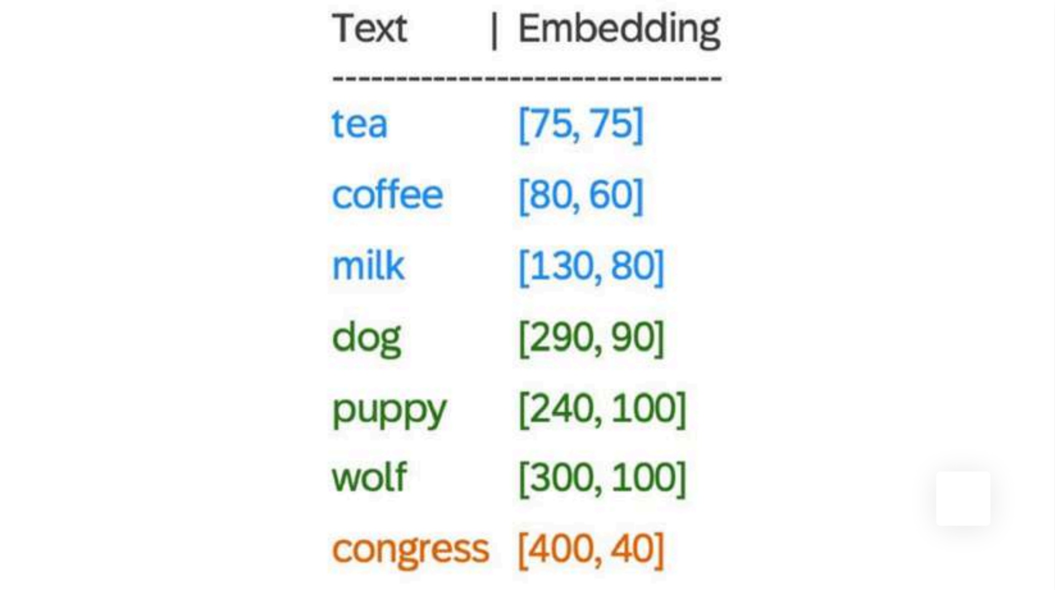
Despite prior advancements, the company still faces significant hurdles in email management that require deeper integration and intelligence:

1. **Structured Grounding of Information:** Ensuring extracted email information is grounded in a structured format for internal applications is critical for accurate categorization and prioritization. For example, a maintenance request needs details anchored within the company's systems. The generative AI hub facilitates this through **document grounding techniques,** leveraging **SAP HANA vector databases** to store and retrieve structured information.
2. **Contextual Understanding with Embeddings:** Making informed decisions requires a deeper understanding of email context and semantics. For instance, recognizing the nuance of customer dissatisfaction is vital for effective resolution. The generative AI hub's **embedding models** can enhance categorization and prioritization by capturing this context, enabling more timely and appropriate responses.

By addressing these evolving challenges, the company aims to streamline email categorization and prioritization further, significantly reduce manual effort, and boost its facility management services' overall efficiency and accuracy. We will see how the generative AI hub can implement these solutions in this unit.

Vector Embeddings

In generative AI, a vector is a mathematical representation that encodes an object's features, typically a list of numbers. An embedding is this vector



Text	Embedding
tea	[75, 75]
coffee	[80, 60]
milk	[130, 80]
dog	[290, 90]
puppy	[240, 100]
wolf	[300, 100]
congress	[400, 40]

Within the **SAP HANA vector engine,** vector embeddings specifically refer to these numerical representations of various data types like text, images, or audio. A **Text Embeddings model** is the model responsible for converting text into these numerical embeddings. These representations are then stored and managed efficiently within the SAP HANA Cloud's vector engine, a key part of its multimodal processing capabilities.

This vector engine enables efficient storage, retrieval, and analysis of high-dimensional vectors. This, in turn, powers advanced applications such as semantic search and Retrieval Augmented Generation (RAG). Integrating vector embeddings with other data types facilitates the development of intelligent data applications and automated decision-making processes.

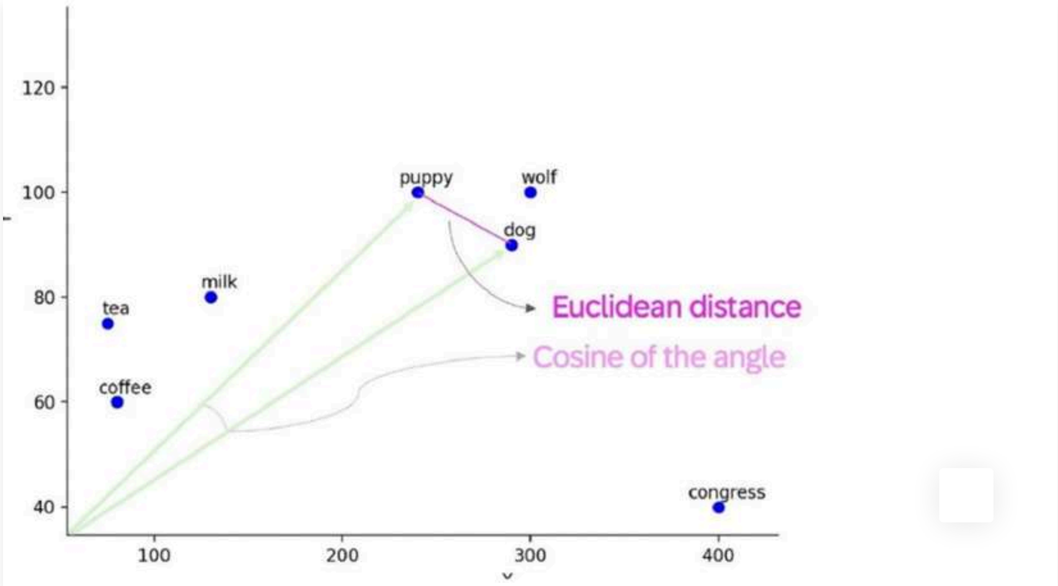
The SAP HANA Vector Engine

The SAP HANA Vector Engine, a core feature of SAP HANA Cloud, empowers storing, processing, and analyzing high-dimensional vectors, such as text embeddings, directly alongside your business data. This engine is integral to SAP HANA Cloud's multimodal processing capabilities, supporting seamless integration with relational, graph, spatial, and document data.

Key functionalities of the SAP HANA Vector Engine include:

- **Efficient Vector Management:** It stores and manages numerical vector embeddings, which represent various data types, such as text, images,

- **Powerful Vector Searches:** The engine supports efficient vector searches directly via SQL, utilizing specialized functions such as L2DISTANCE() and COSINE_SIMILARITY().
    - **L2DISTANCE():** This function calculates the Euclidean distance between two vectors, commonly used to measure straight-line distance in high-dimensional spaces for tasks like clustering and nearest neighbor searches.
    - **COSINE_SIMILARITY():** This function computes the cosine of the angle between two vectors, indicating their directional similarity. It is widely used in text analysis and information retrieval to determine the semantic similarity between documents or embeddings.

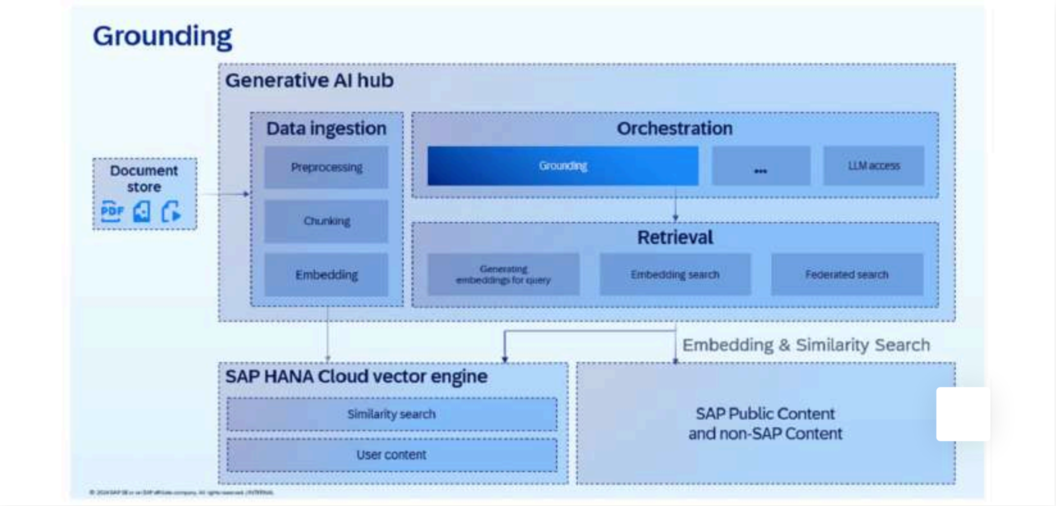


These functions facilitate highly efficient vector searches and can be seamlessly integrated with other SQL operations within SAP HANA Cloud. The engine facilitates context-aware responses and automated decision-making by applying the semantic meaning captured in vector representations.

Overall, the SAP HANA vector engine enhances the ability of intelligent data applications to provide detailed, context-aware responses, significantly improving the overall efficiency and scalability of data processing within SAP HANA Cloud.

Document Grounding within the Generative AI Hub

knowledge sources (such as HR policy manuals) to directly supplement the LLM's internal knowledge base, rendering the model more accurate and reliable.



Within the SAP Generative AI Hub, embedding models play a foundational and distinct role in enabling this grounding process. Unlike generative models, embedding models themselves do not produce text or answers. Instead, they are specialized AI components, like the Text Embedding model, designed to convert raw data (such as document chunks or user queries) into numerical vector representations. These vectors capture the semantic meaning of the data, making it computationally understandable and comparable. This capability is crucial for implementing RAG and enhancing AI responses' contextual relevance and accuracy.

The system is built upon several interconnected key components:

- Document Store: This centralized knowledge base houses various document types (e.g., PDFs and text files).
- Generative AI Hub: Acting as the central coordinator, this hub manages the entire information processing and retrieval process. It orchestrates three primary stages:
    - Data Ingestion: Incoming documents are preprocessed, segmented into smaller chunks, and converted into numerical vector embeddings using embedding models.
    - Orchestration: This stage manages the overall workflow, including the grounding process and interaction with LLMs.

- SAP HANA Cloud Vector Engine: This component provides the fundamental infrastructure for conducting similarity searches based on the generated embeddings, effectively storing user-provided content.
- Embedding & Similarity Search: This critical function connects the retrieval stage with SAP and non-SAP content sources, matching information based on the similarity between query and document embeddings.

Essentially, the system ingests documents, transforms them into a searchable format, and then intelligently uses embedding and federated search techniques to retrieve the most relevant information based on user queries. This finally facilitates a more robust and grounded text generation by LLMs.

Lesson Summary

This lesson introduced document grounding as an essential technique for enhancing Generative AI responses by connecting LLMs with external, trusted knowledge sources. We explored a facility management scenario to illustrate the business challenges addressed by grounding, then delved into foundational concepts like vector embeddings and the SAP HANA vector engine. You learned how the generative AI hub orchestrates this process, leveraging these components to deliver accurate, contextually relevant, and reliable AI outputs, preparing you for practical implementation.

Next lesson

Was this lesson helpful? ☺ Yes ☹ No

Learning

Quick links

Download Catalog (CSV, JSON, XLSX, XML)

SAP Community

Newsletter

Learning Support

Get Support

Share Feedback

Release Notes

About SAP

Company Information

Copyright

Trademark

Worldwide Directory

Careers

News and Press

Site Information

Privacy

Terms of Use

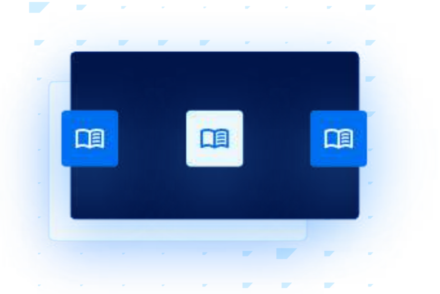
Legal Disclosure

Do Not Share/Sell My Personal Information (US Learners Only)

Preferências de Cookies

Free COURSE • 40 min

Using Advanced AI Techniques with SAP's Generative AI Hub



Start learning

Learning outcome

By completing this course,

- You will understand the principles of document grounding and how vector embeddings enable semantic search and Retrieval Augmented Generation in SAP HANA Cloud.
- You will learn to create and manage a vector knowledge base, configure the Document Grounding module in the Orchestration Service, and generate contextually relevant AI responses using your...

See More

Course information

What you'll learn

UNIT 1

Mastering Document Grounding Using Generative AI Hub

2 Lessons 40 min

Level up your skills

Subscription

Optional but recommended to boost your expertise.

LIVE SESSION

Generative AI Agents

LIVE SESSION

Build Your Own AI Solution with Generative AI hub

LIVE SESSION

Getting Started with SAP Generative AI Launchpad

LIVE SESSION

Get Certified: SAP Certified - SAP Generative AI Developer

SAP Learning

Quick links

- Download Catalog (CSV, JSON, XLSX, XML)
- SAP Learning Hub
- SAP Training Shop
- SAP Developer Center
- SAP Community
- Newsletter

Learning Support

- Get Support
- Share Feedback
- Release Notes

About SAP

- Company Information

Knowledge quiz

It's time to put what you've learned to the test, get 2 right to pass this unit.

1. What is the primary purpose of evaluating Large Language Models (LLMs) in the context of the SAP Cloud SDK for AI?

Choose the correct answer.

○ To determine which model is the cheapest.

● To systematically compare model performance to make an informed selection for business needs.

○ To focus solely on prompt creation efficiency.

○ To eliminate the need for multiple models in all scenarios.

Correct

Systemically comparing the performance of different LLMs is crucial for making an informed, business-driven selection that meets specific requirements.

○ It guarantees high performance across all tasks.

○ It offers the lowest cost solution available.

● It introduces risk due to lack of redundancy and reliability.

○ It simplifies the evaluation process.

Correct

Relying on a single model can introduce risk; evaluating multiple LLMs provides tested alternatives, enhancing solution robustness and minimizing downtime.

YOUR SCORE

2/2

You passed Evaluating Prompts Using Multiple Models

1 Lesson 20min

Next up: Getting Started with Generative AI Hub

Try Again

Selecting a Suitable Large Language Model

Objective

After completing this lesson, you will be able to evaluate different models using SAP Cloud SDK for AI.

Multiple Models in generative AI hub

We've refined our prompts and even enhanced them with multi-modal input. Now, it's time to choose the best LLM to power our solution. In this lesson, we'll move beyond evaluating just prompts to evaluating the **models themselves.** You'll learn how to systematically test various LLMs available in the generative AI hub using the **SAP Cloud SDK for AI** comparing their results on our Facility Solutions Company scenario to make an informed, business-driven selection.

Let's now evaluate different models for Facility solutions problem that we're solving.

Solution Using Different Models

Continuing with our scenario, we've learned to create and refine prompts that assign urgency, sentiment, and categories to customer messages. We also evaluated these advanced prompting techniques to analyze their results.

We will now see how these prompts perform with **different LLMs** available through the generative AI hub. This is important in building a robust solution because the choice of LLM significantly impacts your application's accuracy, efficiency, and overall cost.

summarization, or even handling multimodal inputs (like text and images).

- **Performance:** Not all models perform equally on every task. Comparing them helps you find the most accurate or suitable LLM for your specific requirements.
- **Cost Efficiency:** You can save cost by choosing an LLM that is just right for your task. Sometimes, a simpler, more affordable model can deliver the necessary accuracy, allowing you to avoid the higher costs associated with much more powerful models when they aren’t strictly needed.
- **Flexibility:** Different LLMs offer varied capabilities, including support for various input types or generating diverse output formats, providing a more comprehensive solution for complex needs.
- **Redundancy and Reliability:** For critical enterprise applications, relying on a single model introduces risk. Evaluating multiple LLMs provides tested alternatives, enhancing your solution’s robustness and minimizing downtime.

Different Models in generative-AI-hub Code

mistralai ai models

We begin with **mistralai ai models** and use the basic prompt. These models are the less expensive open source, SAP hosted models available on generative AI hub.

Python

```python
overall_result["basic--mixtral-large-instruct"]
pretty_print_table(overall_result)

```

This code evaluates a dataset and prints the results. It calculates a specific model's performance on a small test set, storing the results under a key in the "overall_result" dictionary. The "pretty_print_table" function then

Code Snippet

```
0%|          | 0/20 [00:00<?, ?it/s]
                                         is_val
===============================================
                     basic--llama3.1-70b
                  few_shot--llama3.1-70b
             metaprompting--llama3.1-70b
metaprompting_and_few_shot--llama3.1-70b
           basic--mixtral-large-instruct
```

Similarly, let's evaluate results using a combination of few-shot and metaprompting for the same model.

Python

```python
overall_result["metaprompting_and_few_shot--mix
pretty_print_table(overall_result)

```

You will see the evaluation results.

Open AI models

We perform similar steps with **Open AI models**. These models are one of the best proprietary OpenAI models available on generative AI hub.

Python

Code Snippet

```
0%|          | 0/20 [00:00<?, ?it/s]

===============================================
                               basic--llama3.1-
                            few_shot--llama3.1-
                       metaprompting--llama3.1-
          metaprompting_and_few_shot--llama3.1-
                     basic--mixtral-large-instr
metaprompting_and_few_shot--mixtral-large-instr
```

Similarly, let's evaluate results using a combination of few-shot and metaprompting for the same model.

Python

```python
overall_result["metaprompting_and_few_shot--gpt
pretty_print_table(overall_result)

```

You will see the evaluation results.

Gemini models

We perform similar steps with **Gemini models**. These models are best Google models available on generative AI hub.

Python

You can have the following output:

Code Snippet

```
 0%|          | 0/20 [00:00<?, ?it/s]

==============================================
                               basic--llama3.1
                            few_shot--llama3.1
                       metaprompting--llama3.1
          metaprompting_and_few_shot--llama3.1
                     basic--mixtral-large-inst
metaprompting_and_few_shot--mixtral-large-inst
                                      basic--g
                 metaprompting_and_few_shot--g
                           basic--gemini-2.5-f
```

You can see results for these outputs.

Similarly, let's evaluate results using a combination of few-shot and metaprompting for the same model.

Python

```python
overall_result["metaprompting_and_few_shot--gem
pretty_print_table(overall_result)

```

You can see the evaluation results.

```
==============================================
                               basic--llama3.1
                            few_shot--llama3.1
                       metaprompting--llama3.1
          metaprompting_and_few_shot--llama3.1
                     basic--mixtral-large-inst
metaprompting_and_few_shot--mixtral-large-inst
                                      basic--g
                 metaprompting_and_few_shot--g
                           basic--gemini-2.5-f
      metaprompting_and_few_shot--gemini-2.5-f
```

Note

You may get a slightly different response to the one shown here and in all the remaining responses of models shown in this learning journey.

When you execute the same prompt in your machine, a LLM produces varying outputs due to its probabilistic nature, temperature setting, and non-deterministic architecture, leading to different responses even with slight setting changes or internal state shifts.

Exercise

In exercises later, you will explore how to select the optimal LLMs for your business needs by leveraging the Model Library in the SAP Generative AI Hub.

Evaluation Guidelines for Different Models

When selecting an LLM within the SAP's generative AI hub, pricing and various factors play a crucial role. Key considerations include:

Orchestration for pricing details in generative AI hub.

- **Scalability:** Consider how easily the model's pricing and infrastructure can scale with your application's growth. Subscription-based models offered in the generative AI hub provide predictable costs and are designed to support scalable AI development and deployment.
- **Performance vs. Cost Balance:** High-performing models typically come at a higher cost. Organizations must evaluate whether the incremental performance gains of a more powerful model truly justify the additional expense for their specific application and its business value. Sometimes, a slightly less performant but significantly cheaper model offers better overall value.
- **Flexibility:** Look for pricing and model options that allow for adjustments based on fluctuating usage patterns or evolving AI demands. This adaptability is crucial for optimizing spending in dynamic enterprise environments.

By considering these guidelines, businesses can make informed decisions about which Generative AI models to deploy, achieving the best balance between cost, performance, and strategic fit for their SAP-integrated solutions.

Evaluation Summary

We saw how the generative AI hub can solve a business problem and learned about its features and options for supporting custom-built AI solutions.

Throughout this course, you've gained a comprehensive understanding of this process. We embarked on an iterative path:

- Starting with basic prompt creation in **SAP AI Launchpad.**
- Scaling our solution by recreating prompts and interactions using the **SAP Cloud SDK for AI.**
- Establishing a baseline through systematic evaluation.
- Enhancing prompt accuracy and effectiveness with advanced techniques like **Few-shot Prompting and Meta-prompting, and even incorporating multi-modal input.**
- Finally, we evaluated various LLMs offered by the generative AI hub, comparing their performance, cost, and suitability for our specific business needs.

other applications within the organization, significantly enhancing customer service and operational efficiency.

The SAP's generative AI hub empowers you to develop, deploy, and manage custom-built AI solutions that enhance your existing business applications programmatically, driving innovation across your enterprise.

Access Models in Generative AI Hub

Continuing with the scenario discussed previously, we created prompts and prompt templates that assign urgency, sentiment, and categories to customer messages that can be used in software.

We used the few-shot technique to arrive at a better prompt.

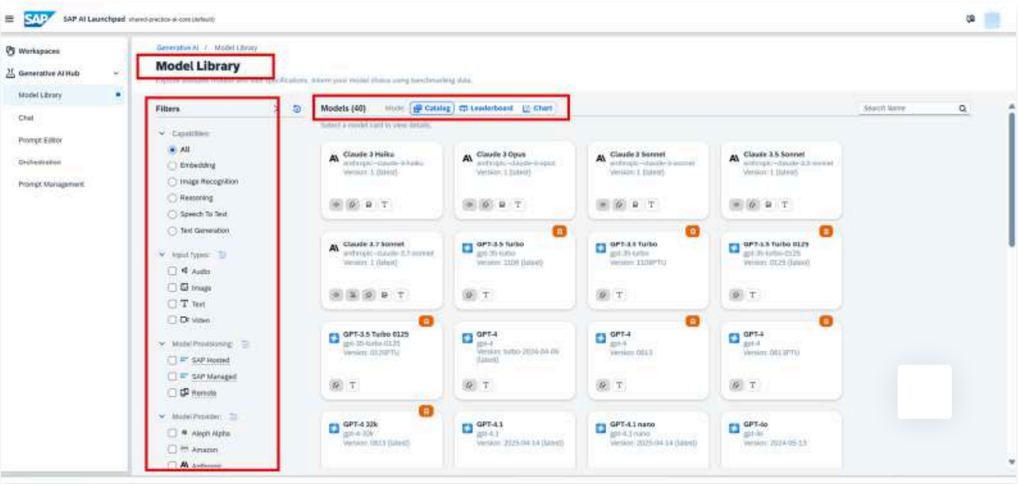
We used prompt template to help scale the solution.

Task 1: Access Different Models using Model Library

We will start with exploring Model Library.

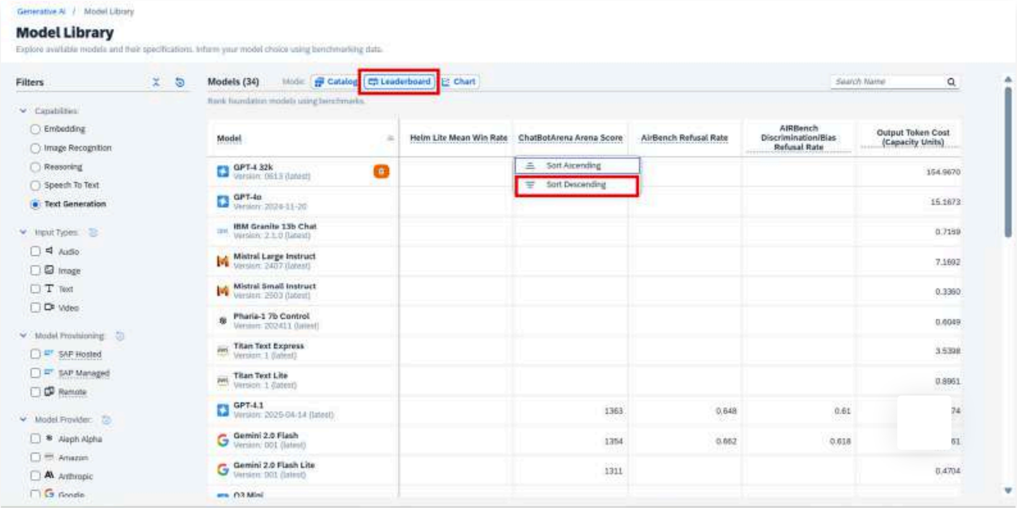
Steps

1. Navigate to **Model Library** in the left pane.
2. You will see the Model Library interface.



The model library provides comprehensive information on models available in the generative AI hub to support informed decision-making. To explore the available models and their metadata, utilize the catalog

3. You can apply **filters** such as capabilities, input types, Model provider etc.
4. Select **Leaderboard.**
5. Select any criteria based on your business needs. For example, select ChatBot Arena score. You can hover over any column to know about them.
6. Select the column and click **Sort Descending.**



Generative AI / Model Library

Model Library

Explore available models and their specifications. Inform your model choice using benchmarking data.

Filters

- Capabilities:
  - ○ Embedding
  - ○ Image Recognition
  - ○ Reasoning
  - ○ Speech To Text
  - ● Text Generation
- Input Types:
  - □ Audio
  - □ Image
  - □ Text
  - □ Video
- Model Provisioning:
  - □ SAP Hosted
  - □ SAP Managed
  - □ Remote
- Model Provider:
  - □ Aleph Alpha
  - □ Amazon
  - □ Anthropic
  - □ Google

Models (34) Mode: Catalog | Leaderboard | Chart

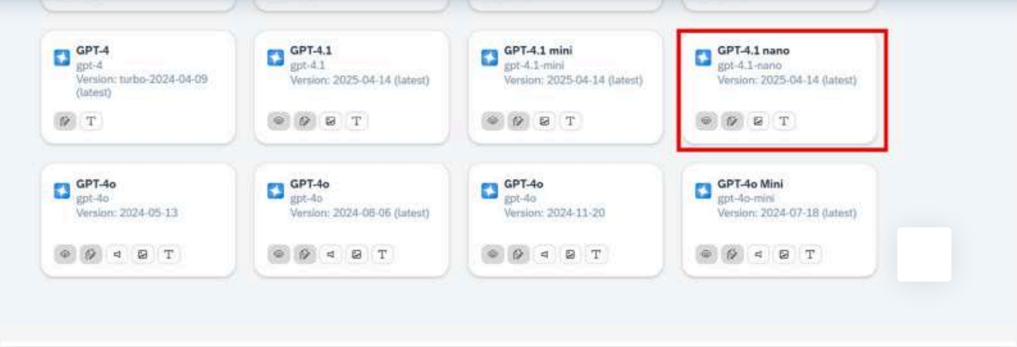
Rank foundation models using benchmarks.

Model	Helm Lite Mean Win Rate	ChatBotArena Arena Score	AirBench Refusal Rate	AIRBench Discrimination/Bias Refusal Rate	Output Token Cost (Capacity Units)
GPT-4 32k Version: 0613 (latest)					154.9670
GPT-4o Version: 2024-11-20					15.1673
IBM Granite 13b Chat Version: 2.1.0 (latest)					0.7159
Mistral Large Instruct Version: 2407 (latest)					7.1692
Mistral Small Instruct Version: 2503 (latest)					0.3360
Pharia-1 7b Control Version: 202411 (latest)					0.6049
Titan Text Express Version: 1 (latest)					3.5398
Titan Text Lite Version: 1 (latest)					0.8961
GPT-4.1 Version: 2025-04-14 (latest)		1363	0.648	0.61	74
Gemini 2.0 Flash Version: 001 (latest)		1354	0.662	0.618	61
Gemini 2.0 Flash Lite Version: 001 (latest)		1311			0.4704
O3 Mini					

You can see **model ratings.** Similarly, you can compare ratings of different benchmarks in the Chart option.

> **Note**
>
> You can see all the models that are offered in generative AI hub. However, this system is configured to allow few selected models only. These are : GP4.1 nano, GPT4o-mini, and Gemini 2.0 Flash Lite, and Mistral Small Instruct

7. Go back to Catalog mode and Search and select GPT 4.1 nano in the Catalog tab.



GPT-4
gpt-4
Version: turbo-2024-04-09 (latest)

GPT-4.1
gpt-4.1
Version: 2025-04-14 (latest)

GPT-4.1 mini
gpt-4.1-mini
Version: 2025-04-14 (latest)

GPT-4.1 nano
gpt-4.1-nano
Version: 2025-04-14 (latest)

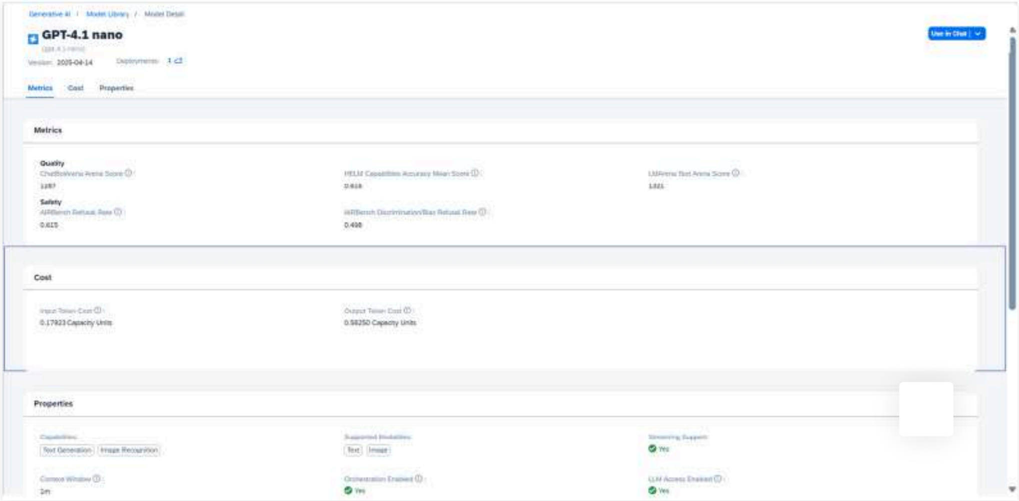
GPT-4o
gpt-4o
Version: 2024-05-13

GPT-4o
gpt-4o
Version: 2024-08-06 (latest)

GPT-4o
gpt-4o
Version: 2024-11-20

GPT-4o Mini
gpt-4o-mini
Version: 2024-07-18 (latest)

8. The model card is displayed. These cards provide all the details about the models in terms Metrics, Cost, and Properties.



GPT-4.1 nano
(gpt-4.1-nano)
Version: 2025-04-14 Deployments: 1

Metrics

Quality
ChatbotArena Arena Score: 1287
HELM Capabilities Accuracy Mean Score: 0.618
LMArena Text Arena Score: 1321

Safety
AIRBench Refusal Rate: 0.615
AIRBench Discrimination/Bias Refusal Rate: 0.498

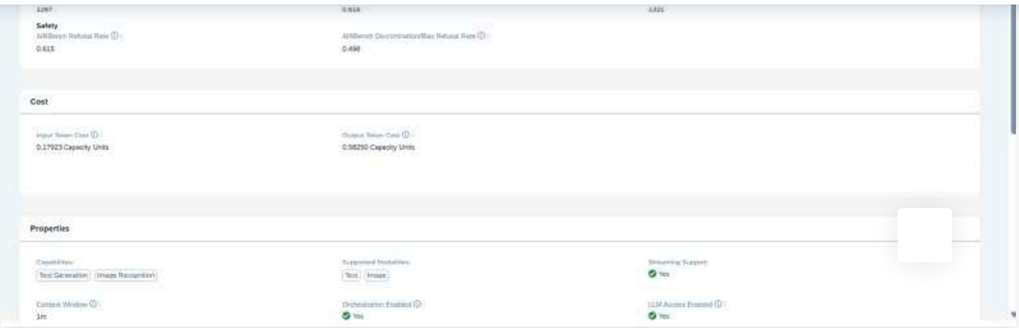
Cost

Input Token Cost: 0.17923 Capacity Units
Output Token Cost: 0.58250 Capacity Units

Properties

Capabilities: Text Generation, Image Recognition
Supported Modalities: Text, Image
Streaming Support: Yes
Context Window: 1m
Orchestration Enabled: Yes
LLM Access Enabled: Yes

9. You can deploy or use the deployed model directly from Model Library. Select the **Use in Chat or Use in Prompt Editor** options based on your need. Here we will select the **Use in Chat option.**



1267 0.618 1321

Safety

AIRBench Refusal Rate ⓘ: 0.615

AIRBench Discrimination/Bias Refusal Rate ⓘ: 0.498

Cost

Input Token Cost ⓘ: 0.17923 Capacity Units

Output Token Cost ⓘ: 0.58250 Capacity Units

Properties

Capabilities: Text Generation, Image Recognition

Supported Modalities: Text, Image

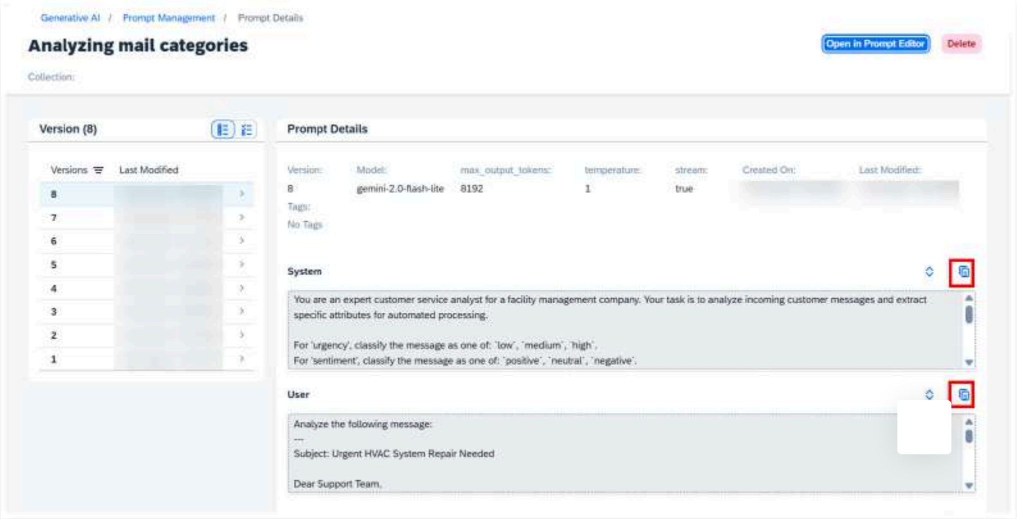
Streaming Support: ✓ Yes

Context Window ⓘ: 1m

Orchestration Enabled ⓘ: ✓ Yes

LLM Access Enabled ⓘ: ✓ Yes

10. Copy and paste your final prompt in the previous exercises and see the results.
11. To copy and paste the prompt, navigate to **Prompt Management, select Prompts,** and then select the **Analyzing mail categories prompt.** Select the latest version and then click **copy.**



Generative AI / Prompt Management / Prompt Details

Analyzing mail categories

Collection:

Version (8)

Versions	Last Modified
8	[redacted]
7	[redacted]
6	[redacted]
5	[redacted]
4	[redacted]
3	[redacted]
2	[redacted]
1	[redacted]

Prompt Details

Version:	Model:	max_output_tokens:	temperature:	stream:	Created On:	Last Modified:
8	gemini-2.0-flash-lite	8192	1	true	[redacted]	[redacted]

Tags:
No Tags

System

You are an expert customer service analyst for a facility management company. Your task is to analyze incoming customer messages and extract specific attributes for automated processing.

For 'urgency', classify the message as one of: `low`, `medium`, `high`.
For 'sentiment', classify the message as one of: `positive`, `neutral`, `negative`.

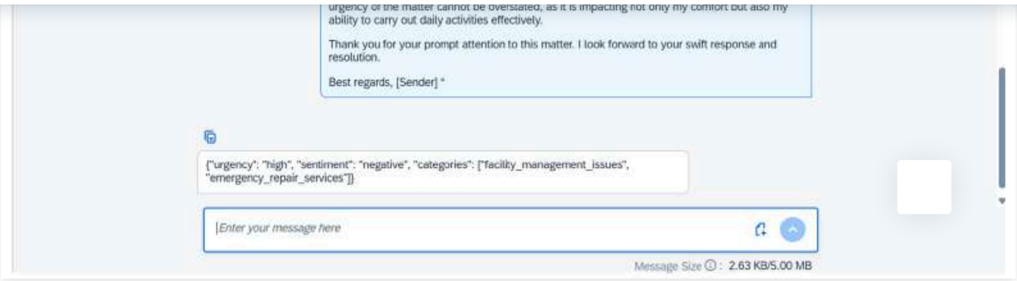
User

Analyze the following message:

Subject: Urgent HVAC System Repair Needed

Dear Support Team,

You can copy messages for each role in a document and use them one by one, taking advantage of the chat interface.



urgency of the matter cannot be overstated, as it is impacting not only my comfort but also my ability to carry out daily activities effectively.

Thank you for your prompt attention to this matter. I look forward to your swift response and resolution.

Best regards, [Sender] "

{"urgency": "high", "sentiment": "negative", "categories": ["facility_management_issues", "emergency_repair_services"]}

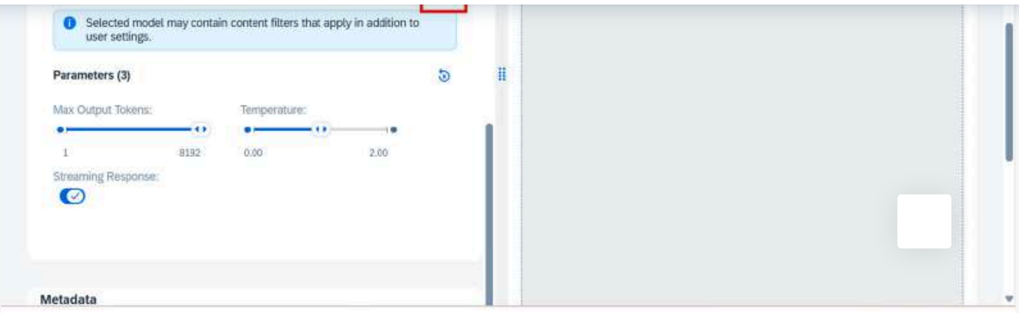
Similarly, you can see results from other models and select the best model for your use case.

Task 2: Access Different Models using Prompt Editor

We will use latest version of the prompt template created in the previous exercises. This is the latest few-shot prompt version with variables and their default values. We will execute this prompt template with different models.

Steps

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 5.0.0.
4. Select the prompt template and then click **Open in Prompt Editor.** Your prompt is ready to use.
5. Scroll to the **Model Configuration tab.**
6. Click **Selected Model.**



Selected model may contain content filters that apply in addition to user settings.

Parameters (3)

Max Output Tokens: 1 8192

Temperature: 0.00 2.00

Streaming Response:

Metadata

7. The Model Selection dialog box is displayed.

8. Select **GPT-4o Mini.**

Model Selection

Learning Browse Get Certified My Learning Subscribe Explore SAP

/ Browse / Courses / Solve your business problems using prompts and LLMs in SAP G...

Knowledge quiz

It's time to put what you've learned to the test, get 3 right to pass this unit.

1. What is the primary advantage of using multi-modal input in generating AI responses?

Choose the correct answer.

● It enhances the clarity of information by combining text and images.

○ It eliminates the need for coding entirely.

○ It guarantees the LLM will always understand user prompts.

○ It simplifies the user interface of the AI systems.

Correct

By integrating both text and images, multi-modal input provides a complete context, allowing the LLM to better understand the problem being described.

methods?

Choose the correct answer.

○ They are always more accurate than larger models.

○ They do not use any examples at all.

○ They are guaranteed to eliminate all errors in processing.

● They require less computational resources and may deliver better results.

Correct

Smaller models and simpler techniques often consume fewer resources and can perform effectively without the overhead of complexity, sometimes providing better results.

3. What is one of the main benefits of combining Few-Shot Prompting with Metaprompting in the context of facility solutions scenario?

Choose the correct answer.

Learning Subscribe

⌂ / Browse / Courses / Solve your business problems using prompts and LLMs in SAP G...

● It allows the model to classify messages with a structured approach while using rich contextual examples.

○ It allows the model to classify tasks for developers using examples.

○ It leads to the creation of less complex instructions for users.

Correct

Combining these techniques utilizes the strengths of context-rich examples and structured guidance, resulting in improved classification accuracy for various tags.

YOUR SCORE

3/3

You passed Refining AI Responses Using Prompt Engineering Techniques

2 Lessons 50min

Next up: Selecting a Suitable Large Language Model

Learning

Subscribe

/ Browse / Courses / Solve your business problems using prompts and LLMs in SAP G...

Learning Browse Get Certified My Learning Subscribe Explore SAP

/ Browse / Courses / Solve your business problems using prompts and LLMs in SAP G...

Enhancing Prompt Effectiveness Through Multi-modal Input

Objective

After completing this lesson, you will be able to optimize AI responses by leveraging multimodal input in your prompts.

Enhancing Prompt Effectiveness Through Multimodal Input

We've mastered the art of refining text-based prompts to get structured information. But what if a picture could make your LLM's job much easier? In the real world, problems often come with visual clues. This lesson will show you how to give your LLM that visual context by using **multi-modal input,** meaning both text and images. We'll explore how to do this directly in SAP AI Launchpad and by extending our code with the **SAP Cloud SDK for AI,** leading to smarter and more accurate AI solutions.

Why Multi-modal Input Matters

Imagine a customer reporting a broken machine. They could describe it in text, but if they also include a photo of the damaged part, the problem becomes much clearer. By allowing your prompts to accept both text and images, you provide the LLM with a complete picture, which can lead to:

- **Better Understanding:** The LLM can "see" what you mean, reducing confusion.

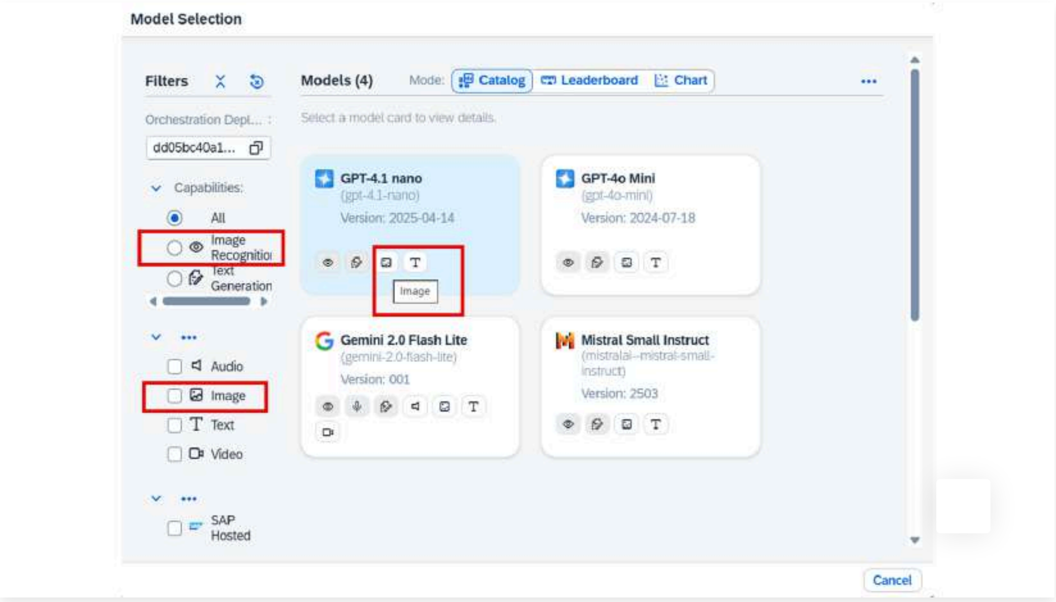
maintenance.

The generative AI hub, with SAP AI Launchpad and the SAP Cloud SDK for AI, makes this powerful capability accessible.

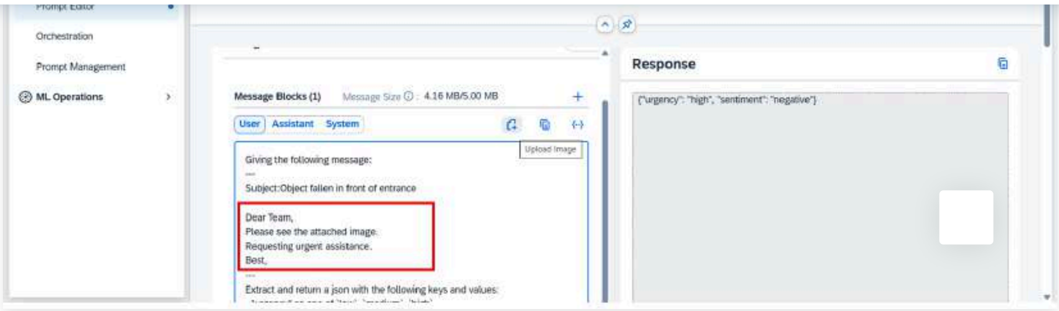
Multi-modal Prompts in SAP Launchpad

You don’t always need to write code to use multi-modal prompts. The SAP AI Launchpad provides a user-friendly interface where you can easily combine text and images. It supports many multi-modal models, such as GPT-4o, allowing you to create and test these advanced prompts visually.

To see which models support multi-modal modes, see Model Library and model cards.



Let’s look at how this appears in the Prompt Editor:



Giving the following message:

Subject:Object fallen in front of entrance

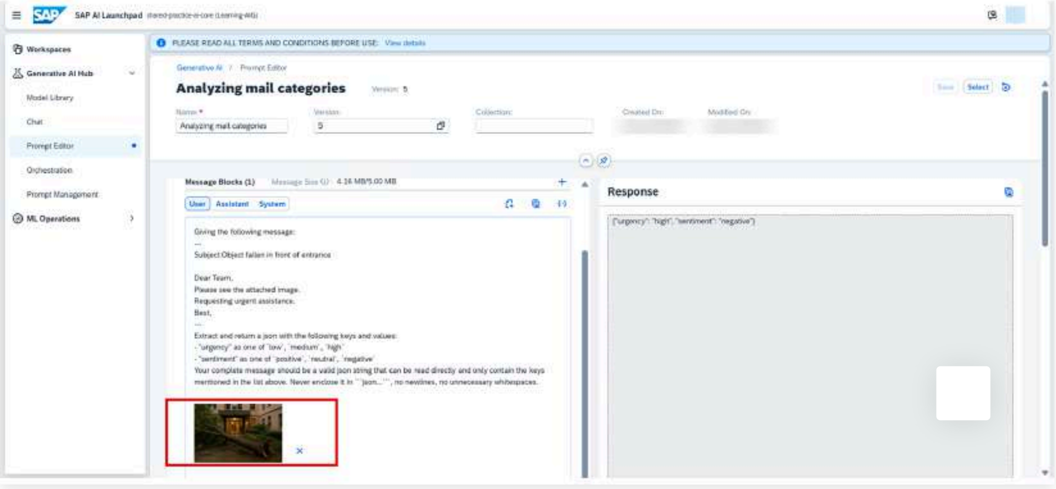
Dear Team,
Please see the attached image.
Requesting urgent assistance.
Best,

Extract and return a json with the following keys and values:

{"urgency": "high", "sentiment": "negative"}

In the Prompt Editor, you've entered a sample email and instructions for the LLM to extract JSON output with urgency and sentiment. You'll notice an **"Upload Image" button.** This is where you can add a visual component to your prompt.

The AI's response for this text-only input might be: {"urgency": "high", "sentiment": "neutral"}.



Analyzing mail categories Version: 5

Name: * Analyzing mail categories
Version: 5
Collection:

Giving the following message:

Subject:Object fallen in front of entrance

Dear Team,
Please see the attached image.
Requesting urgent assistance.
Best,

Extract and return a json with the following keys and values:
- "urgency" as one of `low`, `medium`, `high`
- "sentiment" as one of `positive`, `neutral`, `negative`
Your complete message should be a valid json string that can be read directly and only contain the keys mentioned in the list above. Never enclose it in ```json...```, no newlines, no unnecessary whitespaces.

{"urgency": "high", "sentiment": "negative"}

Here, after clicking "Upload Image" (or dragging and dropping), a small, embedded image is now visible directly within the input text area. This image, showing a fallen tree in front of entrance, is now part of the prompt.

You can see that the complaint text is really short, but with this additional visual information, the AI can often provide a more precise response.

Multi-modal Prompts with the SAP Cloud SDK for AI

For programmatic access and integrating multi-modal capabilities into your custom applications, you can use the SAP Cloud SDK for AI. The main (prompt_13_multimodal) to include an image:

Python

```python

# We need to import TextContent and ImageUr
from gen_ai_hub.orchestration.models.messag
from gen_ai_hub.orchestration.models.templa
from gen_ai_hub.orchestration.service impor
from functools import partial # Imported fo

# The send_request function is updated to a
def send_request(prompt: str, _print: bool 
    # We create a list to hold all parts of
    content_parts = []

    # If an image URL is provided, we add i
    if image_url:
        content_parts.append(ImageUrlConten

    # We always add the text prompt as a Te
    content_parts.append(TextContent(text=p

    # Now, our OrchestrationConfig uses a U
    config = OrchestrationConfig(
        llm=LLM(name=_model),
        template=Template(messages=[UserMes
    )
```

Understanding Code Changes

1. **New Imports:** We now import TextContent and ImageUrlContent from gen_ai_hub.orchestration.models.message. These special types tell the SDK that we're sending different kinds of content.
2. **send_request Update:**

   - The original text prompt is also added to content_parts using TextContent(text=prompt).
   - The UserMessage in our OrchestrationConfig now receives this content_parts list. This tells the SDK to send both the image and the text together to the LLM.

3. **prompt_13_multimodal:** The text of the prompt itself is updated slightly to explicitly tell the LLM to "classify messages and the provided image" and to consider "the image for visual context." This helps guide the LLM's attention.
4. **Calling the Function:** When we create f_13_multimodal using partial, we simply include image_url=example_image_url as an argument. Now, every time f_13_multimodal is called, it will send the text from mail["message"] along with the image at example_image_url to the LLM.

Evaluating Multi-modal Responses

Just like with text-only prompts, it's vital to evaluate multi-modal responses. You'll use the same evaluation functions used previously to check if the JSON output is correctly formatted and if the extracted categories, sentiment, and urgency are accurate. The key difference is that now the LLM has more information (the image) to arrive at its answer, so your "ground truth" for what's correct implicitly includes that visual context. This helps you confirm that adding images genuinely improves your AI's understanding and accuracy.

Practical application

A practical application of this multi-modal capability of generative AI hub can be a web-based intelligent chatbot capable of interacting with users via text, audio, images, and video. It returns context-aware responses using a multi-modal AI model.

See Multimodal Response Assistant Chatbot Using SAP AI Core

Lesson Summary

In this lesson, you've taken a significant step forward by learning to incorporate **multi-modal input** into your generative AI applications. Whether using the intuitive **SAP AI Launchpad** or programmatically with the **SAP Cloud SDK for AI,** you now understand how to provide LLMs with both text and image context. This powerful approach leads to more

Learning

Subscribe

⌂ / Browse / Courses / Solve your business problems using prompts and LLMs in SAP G...

SAP Learning Hub

SAP Training Shop

SAP Developer Center

SAP Community

Newsletter

Learning Support

Get Support

Share Feedback

Release Notes

About SAP

Company Information

Copyright

Trademark

Worldwide Directory

Careers

News and Press

Site Information

Privacy

Terms of Use

Legal Disclosure

Do Not Share/Sell My Personal Information (US Learners Only)

Preferências de Cookies