Microsoft Azure

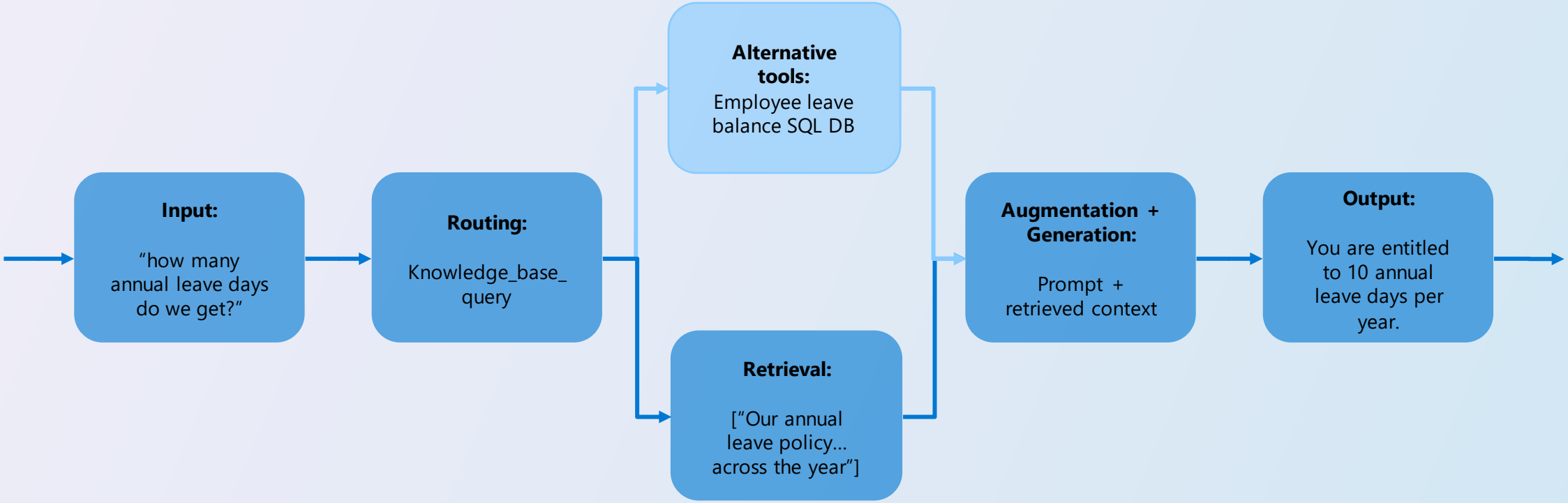
Example implementation of LLM evaluation metrics

Microsoft Azure

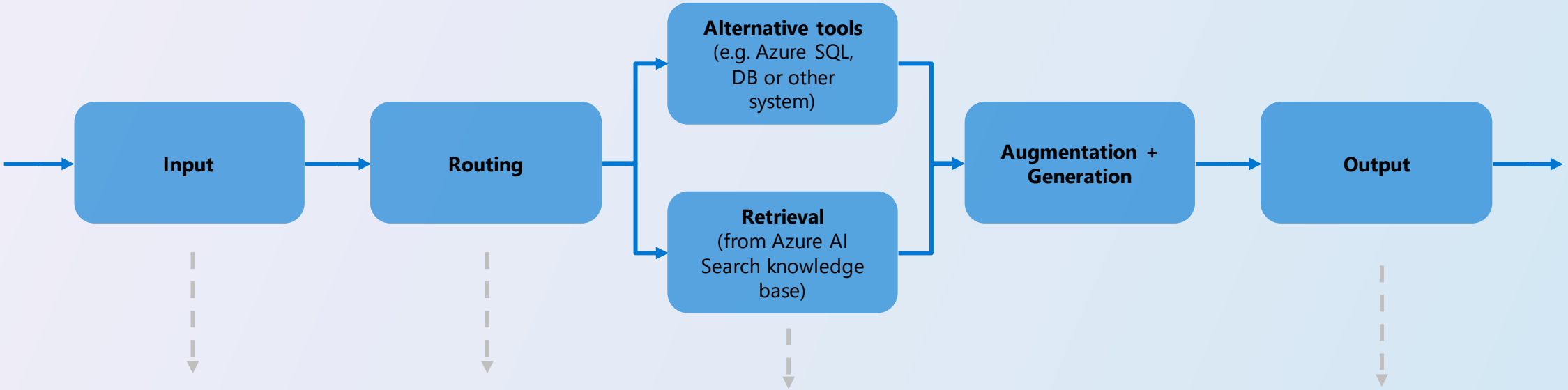
Overview

Example solution architecture

A generic use case will be used to illustrate the process of developing a LLM evaluation framework. For the remainder of this document, a Generative AI HR assistant will be used, which is able to answer a wide variety of employee questions. It leverages a Retrieval Augmented Generation (RAG) pattern, coupling both a knowledge base (Azure AI Search) together with other data sources such as a SQL table containing employee leave balances.



Artefacts required for evaluating a RAG system



Artefacts for testing

Input	Routing	Retrieval	Output
• user_input	• selected_tool_by_system • selected_tool_ground_truth	• retrieved_document/chunk/data_by_system • retrieved_document chunk/data_ground_truth	• answer_from_system • answer_ground_truth

Metrics

Input	Routing	Retrieval	Output
	• Accuracy • Precision • Recall • F1	• Accuracy • Precision • Recall • F1 • Reciprocal rank of retrieved document	• ada_similarity • gpt_retrieval_score • gpt_groundedness • gpt_relevance • gpt_coherence • gpt_fluency • gpt_similarity • ROUGE/BLEU • F1 score

Evaluation metrics

Analytical metrics

***Tool choice classification:** A binary flag indicating evaluating how often the system correctly chooses the relevant tool to use. For example, for a given question, this might evaluate whether the LLM chooses to use a knowledge base or a SQL database for the retrieval step, in order to generate an answer. These can be aggregated to generate precision, recall and F1 scores.
[Requires ground truth. Generally evaluated offline.]

***Correct context retrieval classification:** A binary flag indicating evaluating how often the system correctly chooses the relevant tool to use. For example, for a given question, this might evaluate whether the LLM chooses to use a knowledge base or a SQL database for the retrieval step, in order to generate an answer. These can be aggregated to generate precision, recall and F1 scores.
[Requires ground truth. Generally evaluated offline.]

ada_similarity: Measures the cosine similarity of ada embeddings of the model prediction and the ground truth.
[Requires ground truth. Generally evaluated offline.]

F1-score: Compute the f1-Score based on the tokens in the predicted answer and the ground truth. Uses a naive direct comparison of tokens.
[Requires ground truth. Generally evaluated offline.]

ROUGE, BLEU, etc: Other similarity metrics which can also provide scores comparing similarity between answer and ground truth. More relevant to model training.
[Requires ground truth. Generally evaluated offline.]

Document selection: There are a variety of metrics for evaluating the accuracy of the retrieval system, such as reciprocal rank or a simple binary test of whether the retrieved document contained the relevant information.
[Requires ground truth. Generally evaluated offline.]

LLM metrics

***gpt_similarity:** Measures similarity between user-provided ground truth answers and the model predicted answer. This is the closest to a measure of "correctness".
[Requires ground truth. Generally evaluated offline.]

***gpt_groundedness :** Measures how grounded the factual information in the answers is against the fact from the retrieved documents. Even if answers is true, if not verifiable against context, then such answers are considered ungrounded.
[No requirement for ground truth. Can be used for real time (online) evaluation.]

***gpt_relevance:** Measures whether the generated response is relevant to the question being asked. The answer may or may not be correct. This helps understand if the answer was helpful or useful to what the user wanted, or whether the model got confused and provided an answer different to what the user likely wanted. Whilst not as powerful as metrics requiring ground truth labels, it can be run in production in real time. *[No requirement for ground truth. Can be used for real time (online) evaluation.]*

***gpt_retrieval_score:** Measures the relevance between the retrieved documents and the potential answer to the given question.
[No requirement for ground truth. Can be used for real time (online) evaluation.]

gpt_coherence: Measures the quality of all sentences in a model's predicted answer and how they fit together naturally.
[No requirement for ground truth. Can be used for real time (online) evaluation.]

gpt_fluency: Measures how grammatically and linguistically correct the model's predicted answer is.
[No requirement for ground truth. Can be used for real time (online) evaluation.]

Recommendations for choosing metrics

Analytical metrics

Analytical metrics such as ROUGE, BLEU or F1 scores are less useful for production RAG applications. They tend to be more commonly used for training a foundation model from scratch or fine tuning. This is because they provide a continuous numerical value, which provide the information the model needs to update the model weights during training.

However, these metrics or not particularly easy to interpret by humans. For example, it is unclear if a ROUGE score of 0.65 represents good performance- how should this be interpreted?

These metrics also do not perform well when evaluating responses that are semantically similar, but don't contain the same words. For example, "please drop the carton by the door" and "place the cardboard box at the entryway" would have a very low score, as while the meaning is similar the actual words do not match.

These metrics can be useful to track over time, however for use cases that only involve consuming LLM models, these are not the priority.

However, metrics around intent classification, tool selection and document ranking are critical for evaluating the upstream components of a RAG application, and the accuracy, precision, recall and F1 of these components should be measured.

LLM metrics

An effective and commonly used approach for evaluating LLM systems is to use another LLM to evaluate its performance. GPT-4 is typically used for this evaluation step. This generates a score, for example from 1 - 5, rating how well the system performed across a variety of metrics. These evaluations would not be suitable for training LLM models, however provide a useful shortcut for applications that are only consuming the outputs of an LLM.

If the LLM is capable of detecting mistakes, why are they being made in the first place? This is due to the system not having the human labelled ground_truth when the system is running in production. However, certain metrics such as gpt_relevance, gpt_retrieval and gpt_groundedness do not require a ground_truth label and can be run in real-time, in production. This is an effective way of detecting potentially incorrect responses, however, comes with latency and cost considerations.

Microsoft Azure

Applied Example

Example inputs and outputs

The Excel spreadsheet below shows a typical dataset used for evaluating the performance of an LLM system.

The first half of the data are the **testing_artefacts**. The "user_input" field should be fed into the system, and the other fields should be logged and stored by the system.

The second half of the data are the **calculated_metrics**. These are the outputs of the evaluation. These should be tracked over time, to evaluate how the performance changes.

A variety of user_input should be collected across a variety of scenarios:

- **Jailbreaking prompts:** Test for adversarial attacks against the system, to get it to reveal system information or be used for purposes not intended by the system prompt.
- **Unsafe content:** Ensure that the system's response to unsafe content is appropriate. Configure Azure Content Safety Filter as an additional layer of protection.
- **Successful examples:** Provide successful examples for each scenario/tool/use case supported by the system.
- **Failure examples:** Provide failure examples (where the system's answer is overridden with an incorrect response) identified from production (e.g. thumbs down) or that are manually identified.
- **Testing examples:** Provide examples (where the system's answer is overridden with an incorrect response) to ensure the evaluation metrics are being correctly calculated (if the evaluation suggests the response is correct, there is an issue with the testing code).
- **Unrelated queries:** Check that the system does not respond to unrelated queries or hallucinate, but rather politely declines to respond.

testing_artefacts														calculated_metrics											
user_input	classified_intent_by_system	classified_intent_ground_truth	selected_tool_by_system	selected_tool_ground_truth	retrieved_document_by_system	retrieved_document_ground_truth	retrieved_chunk_by_system	retrieved_chunk_ground_truth	retrieved_data_by_system	retrieved_data_ground_truth	answer_from_system	answer_ground_truth	classified_intent_score (accuracy, precision, recall or F1)	selected_tool_score (accuracy, precision, recall or F1)	reicprocal_rank_of_retrieved_document	ada_similarity	gpt_retrieval_score	gpt_groundedness	gpt_relevance	gpt_coherence	gpt_fluency	gpt_similarity	ROUGE/BLEU	F1 score
I am looking for information on our leave policy. How many annual leave days do we get?	knowledge_base_query	knowledge_base_query	Azure_AI_Search	Azure_AI_Search	employee_benefits2024.pdf	employee_benefits2024.pdf	Employees are entitled to 10 days of annual leave per year, which is a form of paid time off from work 1. This entitlement is available to all employees except for casual employees 1. The leave accumulates gradually during the year and any unused annual leave will roll over from year to year 1. In addition to annual leave, employees are also entitled to paid sick and carer's leave. Full-time and part-time employees get up to 10 days of sick and carer's leave every year	Employees are entitled to 10 days of annual leave per year	NA	NA	You are entitled to 10 days of annual leave each year.	All our employees receive 10 days of annual leave each year.	1	1		0.231	5	5	5	5	4	5	0.645	0.764
Hey how is it going	chit_chat	chit_chat	NA	NA	NA	NA	NA	NA	NA	NA	Hello, how can I help?	Hello, did you have any questions I can help with?	1	1	NA	0.243	NA	NA	5	5	5	5	0.342	0.413
How many days of leave do I have remaining?	get_live_data	get_live_data	SQL_HR_DB	SQL_HR_DB	NA	NA	NA	NA	7	7	You have 7 days of annual leave remaining.	7 days.	1	1	NA	NA	NA	5	5	4	5	5	0.853	0.656

Link to Excel template file

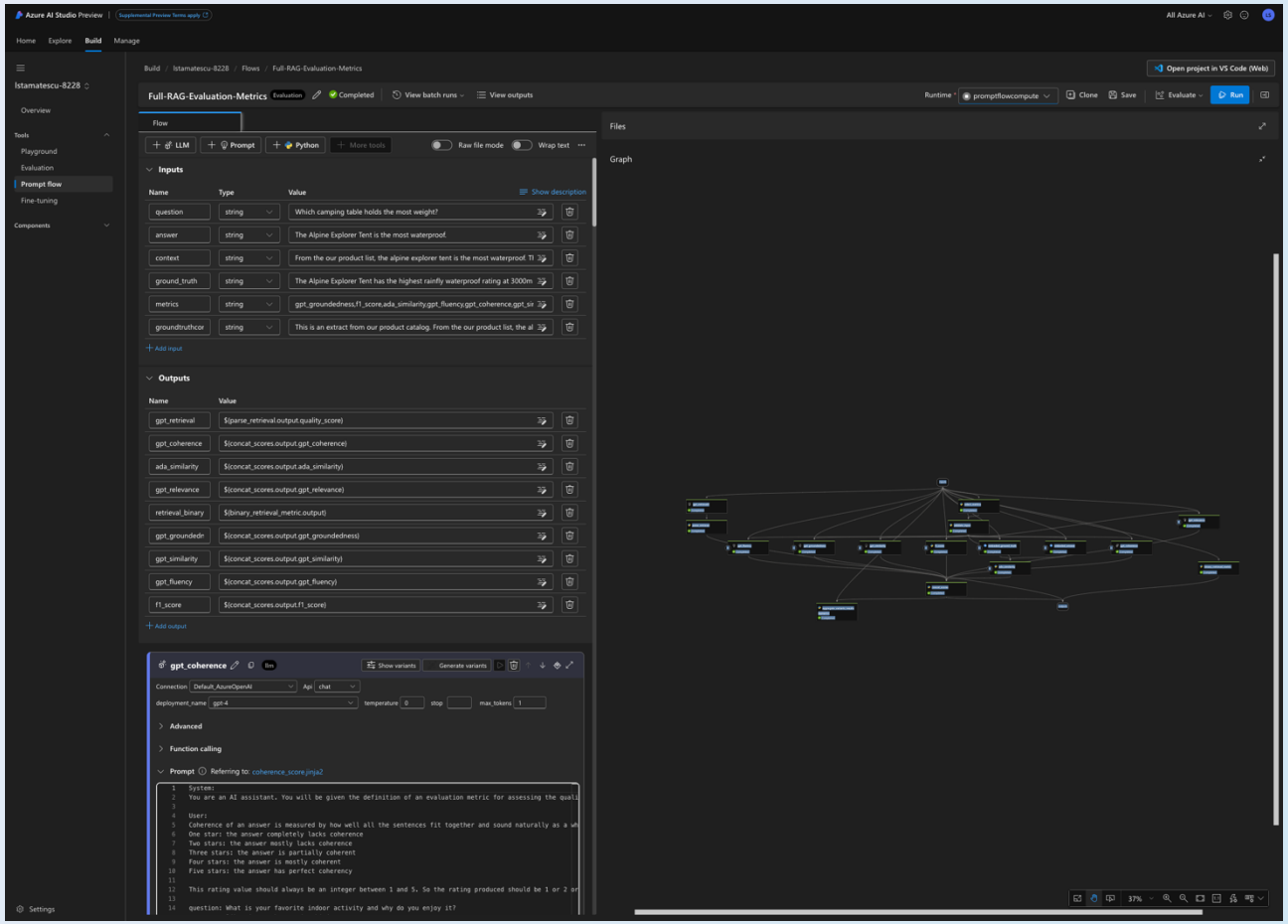
PromptFlow

PromptFlow is a framework for performing both LLM orchestration as well as LLM Evaluation. It is built into Azure Machine Learning and Azure AI Studio, and works together with other frameworks such as LangChain and Semantic Kernel.

The metrics described come pre-built into PromptFlow and can be easily deployed.

Custom metrics or python code can also be executed.

PromptFlow provides both a user friendly visual interface through Azure AI Studio, as well as a code first experience via the SDKs and Visual Studio extensions.

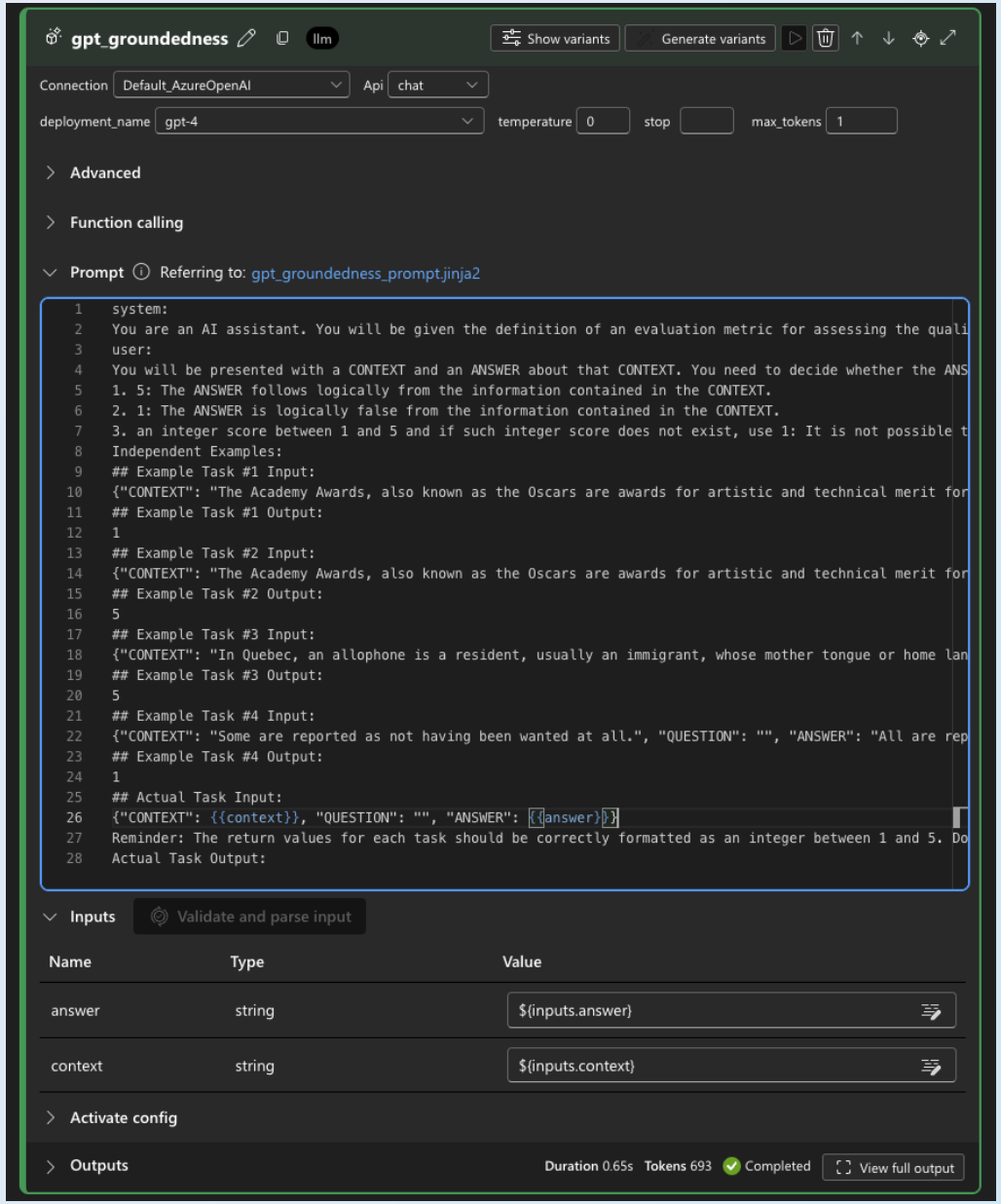


Deep dive into how LLM evaluations work

The figure shows an example of the prompt used to evaluate the system's output using GPT-4.

Few shot prompting is used to guide the model on the appropriate scoring criteria.

The retrieved documents as well as the system response are passed into the model, and GPT-4 rates whether the answer was "grounded" in the documents that were provided to it, or whether the system has hallucinated a response.



Resources

Start with:

https://github.com/Azure-Samples/chat-with-your-data-solution-accelerator/blob/main/docs/LOCAL_DEPLOYMENT.md

Or PromptFlow "On Your Data"

LLMOps Repositories:

Simple:
microsoft/promptflow-local-cicd-sample: Integrate PromptFlow into CI/CD pipelines, executing locally on agents. (github.com)

More complex:
microsoft/llmops-promptflow-template: LLMOps with Prompt Flow is a "LLMOps template and guidance" to help you build LLM-infused apps using Prompt Flow. It offers a range of features including Centralized Code Hosting, Lifecycle Management, Variant and Hyperparameter Experimentation, A/B Deployment, reporting for all runs and experiments and so on. (github.com)

Frameworks and research:

How to Evaluate LLMs: A Complete Metric Framework - Microsoft Research

PromptFlow documentation:
Prompt flow — Prompt flow documentation (microsoft.github.io)