Paper ML-22

Three Pointers for Effective and Accurate LLM Integration

Sundaresh Sankaran, SAS Institute, Cary, NC, USA
Sherrine Eid, SAS Institute, Cary, NC, USA

ABSTRACT

Life Sciences organisations look to integrate Large Language Models (LLMs) in business and analytical processes. We offer guidelines and considerations to ensure LLMs provide you effective, accurate and usable insights that increase your organization's productivity and enriches user experience.

Our suggestions cover three broad areas – one - access and integration of external or hosted LLM services with your analytics; two - design considerations for right parameters and instructions; and three - robust approaches based on Retrieval Augmented Generation (RAG), which improve accuracy and increase trust in LLM results.

We detail how hybrid approaches combining traditional Machine Learning methods, Natural Language Processing and Decisioning capabilities enhance LLM quality manifold, while ensuring efficient cost of operations. Our objective is that this session provides you actionable suggestions and guidelines which are practical and easily implementable, ensuring your LLM projects deliver value to their fullest potential.

INTRODUCTION

After an initial period of experimentation motivated by the buzz around Generative AI (Gen AI) and Large Language Models (LLMs), the Life Sciences industry has focused its efforts around tangible and practical projects which use LLMs to facilitate rapid clinical development, carry out better safety reviews, improve product quality and ensure better patient outcomes.

However, implementing LLMs as an integral part of your life sciences analytical platforms is not a straightforward endeavour. Major challenges to harnessing LLMs and taking full advantage of them include their special infrastructure and resource needs, their deployment mechanisms, their opacity and importantly, inherent risks which hinder trust in AI results.

Compared to early 2024 (when the buzz about LLMs was at its peak), organizations today adopt a more rigorous posture in terms of evaluating LLM opportunities. Note also that life sciences organisations already have existing AI investments that they wish to continue harvesting and enhancing, rather than replace them with LLMs and allied technologies. We present three main considerations to help you make informed decisions on how to incorporate and integrate LLMs into your AI and analytics platforms.

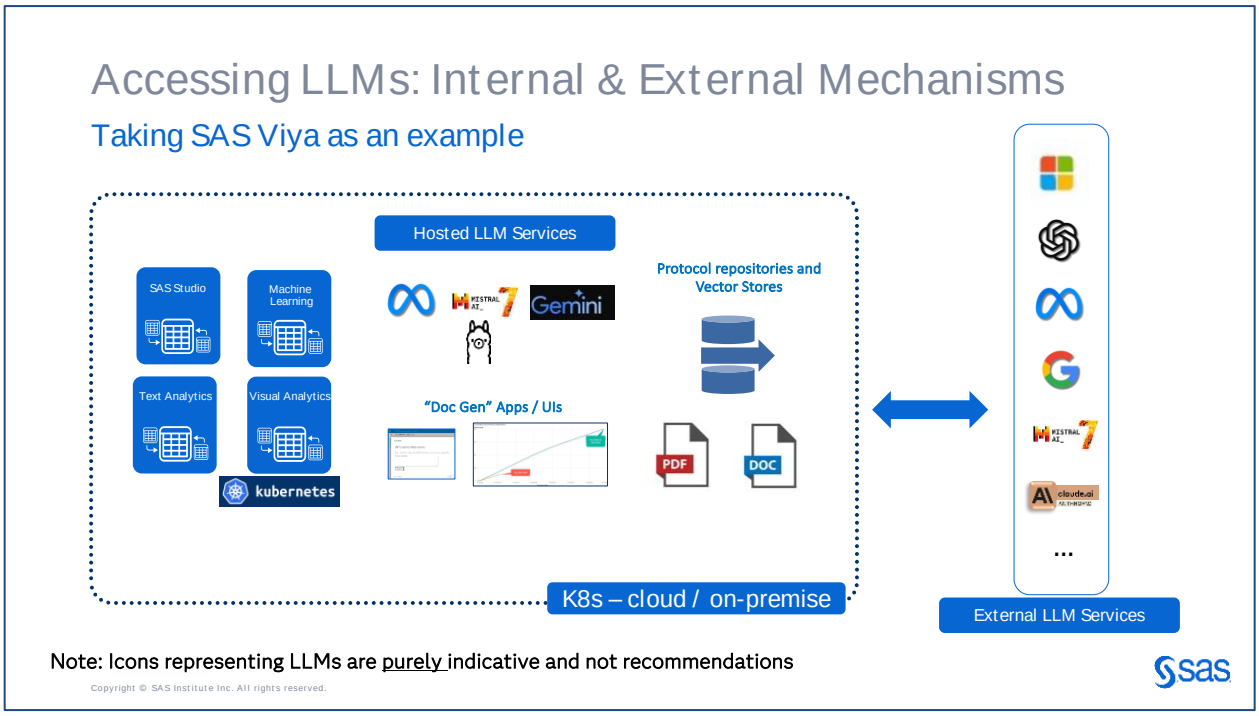
HOW DO YOU ACCESS YOUR LLM?

Large Language Models are delivered through two main mechanisms.

1. **As a Service**: This is usually the most common mechanism through which the lay user encounters an LLM. LLM vendors host their LLMs on a cloud server and expose the same to users as a service. Examples include OpenAI, Azure OpenAI, AWS Bedrock, Claude, and Llama models served through Meta AI. Vendors provide this model to offer customers (life sciences organisations) simplicity and convenience, as well as a standard pricing structure usually based on a subscription model or on transaction size (e.g. number of tokens). Also, note the vendors include both providers of the foundation models (LLMs) as well as service providers who host the foundation LLMs on their platforms. An additional benefit is that the API calls standardise request patterns and schema transmitted to multiple LLMs, reducing the need for the end user to be aware of multiple formats.

2. **As a Package**: LLMs comprise of binary files containing model weights, structure and instruction sets. Some vendors make these LLMs available for download from their websites, which customer organisations can choose to host on their internal servers either with or without modification or customisation. This route is popular for models released under an open-source license. Organisations sometime prefer this method because it provides customisability and allows them to protect their data which they can now send to the in-house / on-premise LLM. Considering the broad-based nature on which most LLMs are trained, organisations also look at this option to fine-tune and customise an LLM in-house and then host it themselves in such a way that this is available only within their internal network and firewalls.

Organisations have different considerations for both approaches. A common, and foremost approach should be to obtain a better awareness of how the LLM processes data and carry out a risk assessment on whether any interaction with the LLM raises the chances of data compromise. The risk assessment may reflect different levels of risk for different types of data and use cases.

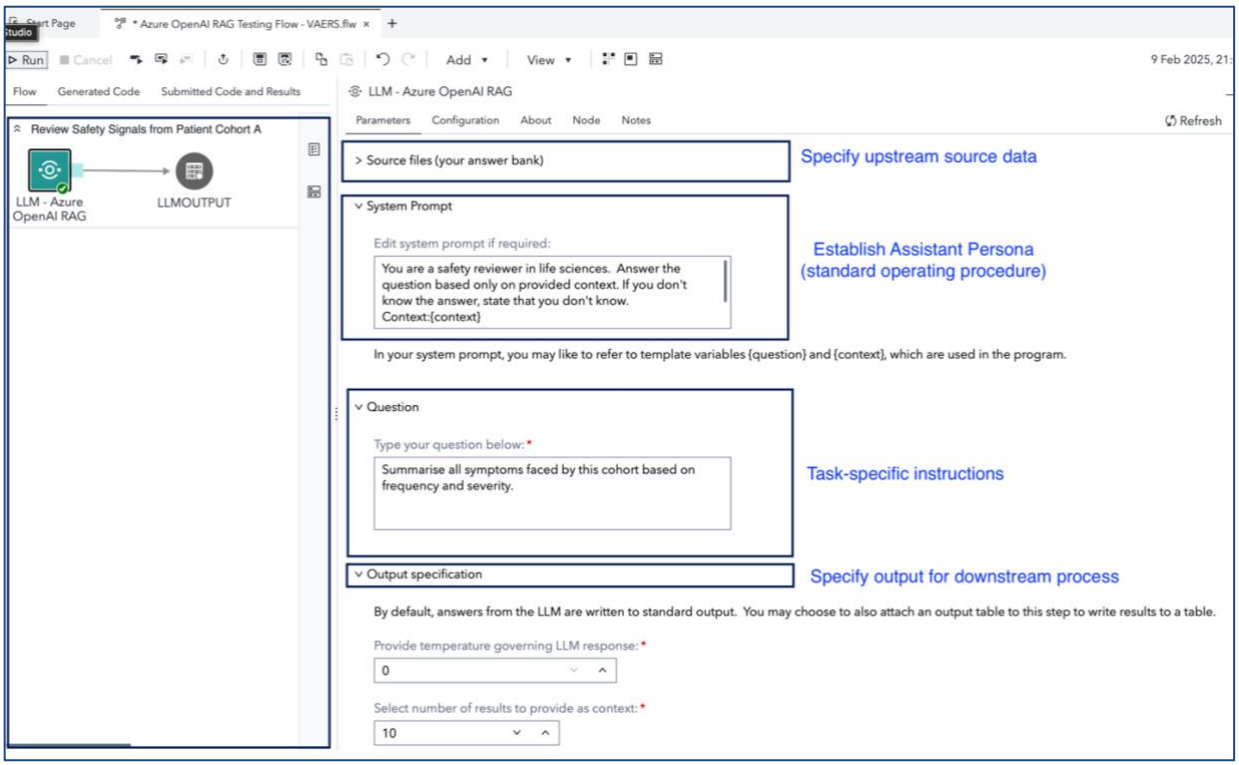


Considerations when integrating with a hosted LLM service should include tools and interfaces which make it easier for the user to interact with the LLM.

LLM interactions aren't standalone, but part of an overall analytical flow which involves data that is sourced from an upstream database, may be filtered for further transformation and then submitted to the LLM in support of a question. Interfaces should accommodate provisions for pointing to the data source and relevant columns which contain the context that is used by the LLM to process the answer. As an example, here's an example of an interface to call an external LLM service (Azure OpenAI) from within the SAS Viya analytical platform:

https://github.com/sassoftware/sas-studio-custom-steps/tree/main/LLM%20-%20Azure%20OpenAI%20RAG

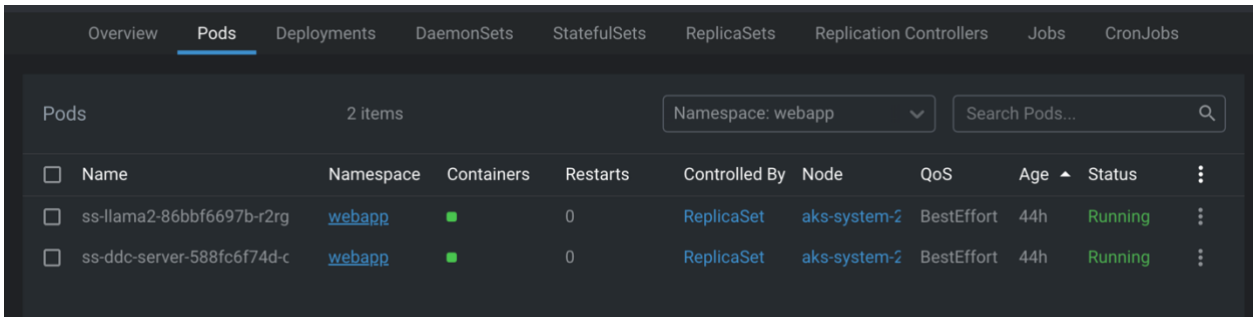
Made available as an open-source contribution on GitHub, this component is a low-code interface meant to assist users to call out to LLMs within a SAS Studio Flow, a processing paradigm for sequential execution of SAS programs as per a defined process. Calling the LLM within a flow as opposed to a disjoint separate activity helps in seamless metadata transfer and full traceability of the interactions, which help in overall process standardisation and impact analysis.

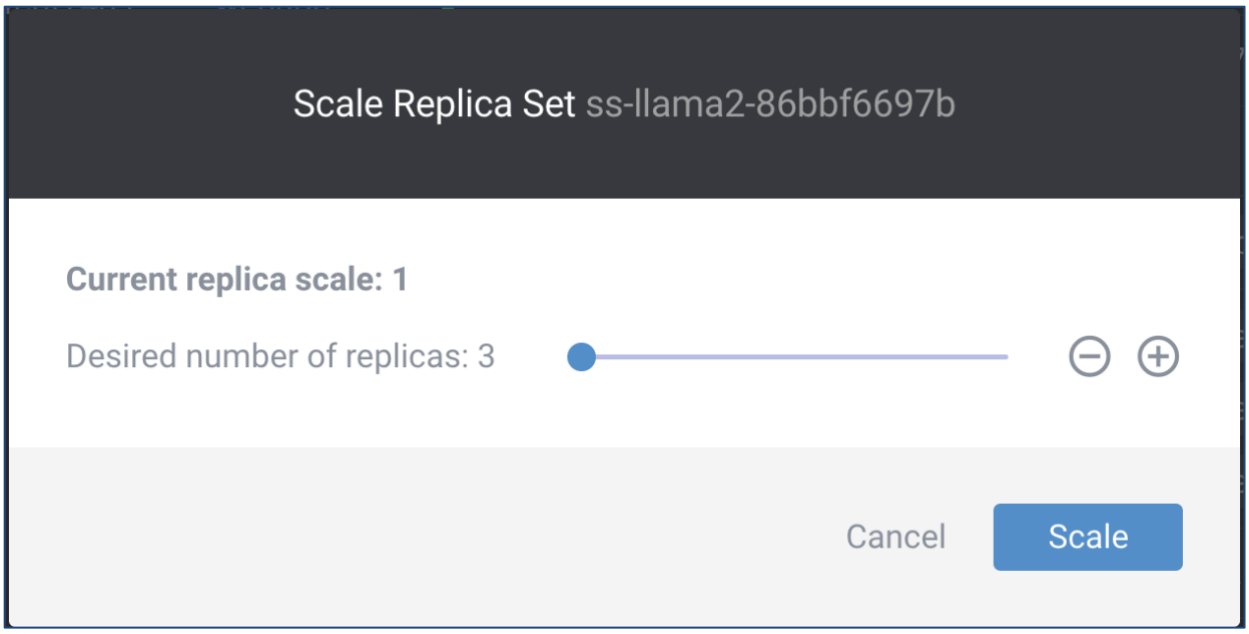


Regarding access to the same capabilities but from an in-house LLM, security considerations tend to predominate when it comes to organisations deciding to host the LLM themselves within their internal networks. Another motivation happens to be the possibility of fine tuning the LLM for specific life-sciences domain and information that the organisation might be interested in.

Upon identifying a list of LLMs that it wishes to make available, organisations need to provide for a service which provides a standard abstraction layer for users to interact with their LLM of choice. For this purpose, several accelerator utilities are available in the market and can be used. An example is Ollama (ollama.com) which can function as a lightweight 'server' for quick access to LLMs.

As an example, a suggested pattern could be to containerise the service and serve it 'alongside' an analytics platform (such as SAS Viya) in a separate namespace within the same Kubernetes cluster.

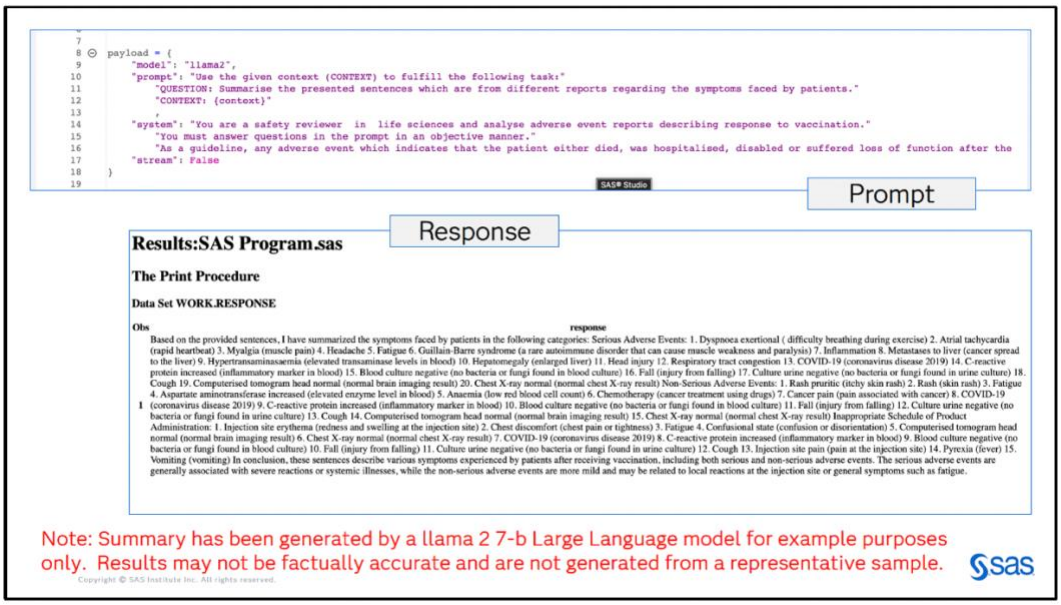




The advantage of hosting your LLM in-house is also that you have the advantage of choosing your infrastructure (which may contain Graphical Processing Units (GPUs) for faster performance), dedicating it for the purpose of serving LLMs and scaling up on demand. In the above screenshot, we deployed a Llama2 model (for example only, your choice of LLM might differ) within a container inside a dedicated node pool. Upon higher concurrency (and correspondingly higher demand for users), it's also possible to scale up the number of replicas which back this service.

The other noteworthy aspect of being able to serve LLMs on-premise is that you would be able to specify tight rules around users or user groups who have access to the service, as identified by IPs / CIDR ranges. This enables organisations to govern access to specific LLMs based on business needs and purposes.

Finally, the adoption of a common service layer through which you access LLMs provides a standard interface and a wrapper around the inner workings of the LLM, which organisations may not wish to expose to users as part of their security posture. Users only need to understand the broad entry level parameters they need to input and provide values and inputs around the same, as the below example shows.

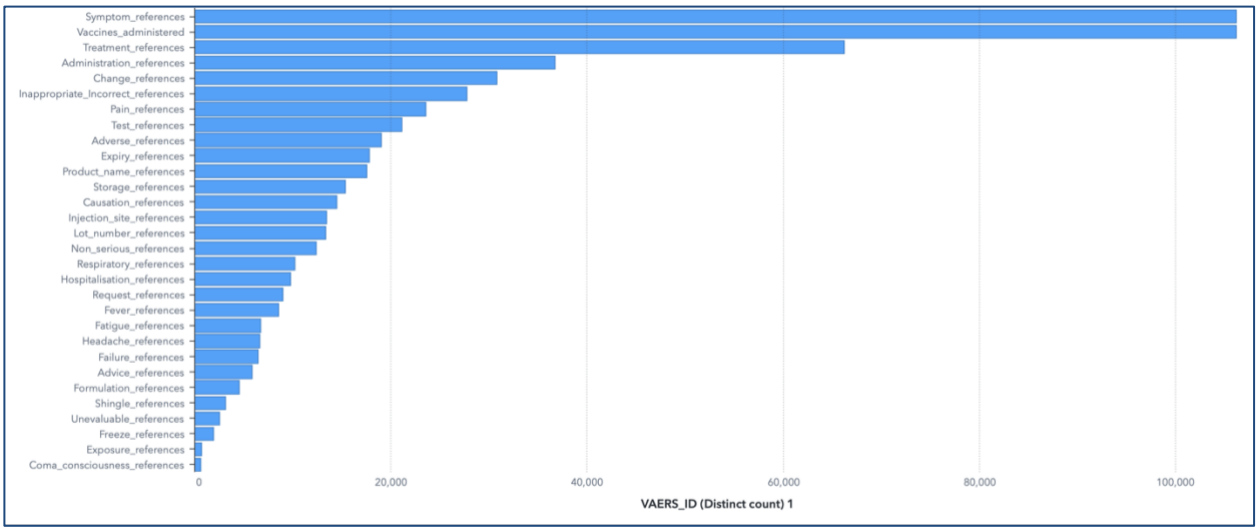


HOW DO YOU INTERACT WITH YOUR LLM?

Users require a paradigm shift when accessing LLMs within the context of their enterprise vs. LLMs for personal purposes / tools. When using LLMs for personal purposes, users tend to ask questions which are of a broader nature, and which fall under the ambit of tasks that the LLM has been pretrained upon.

Enterprise settings, however, require additional context for accurate results. This context comes from within data sources (which may be a composite of unstructured and structured data sources) within the enterprise and may be unique to that organisation. Furthermore, the format and structure in which data currently exists may not be directly usable by the LLM. Current data may be voluminous and unclassified at present and may contain noise which is irrelevant to the task at hand. Therefore, data preparation and transformation activities attain higher importance and prominence when interacting with LLMs in an enterprise scenario.

Data preparation for unstructured data requires a focussed set of capabilities under Natural Language Processing (NLP) and further downstream, in Machine Learning domains. NLP consists of core operations which make sure that the right data is available for the LLM to perform its task accurately. This is carried out through content categorisation, which tags relevant sections of content under appropriate headings so that they can be used appropriately based on the prompt. For example, sections relating to the administration of drugs may be tagged under a heading of administration references, which are then treated as separate extracts under the main subject matter.



The other core operation performed by NLP is standardisation. Unstructured data may represent the same facts using a variety of terms. A simple example would be the use of terms pyrexia or high temperature for fever. Conceptual definition (also known as information extraction) is an NLP capability that helps identify and define multiple variations of the same term under a common heading.

We require efficient and fast retrieval of relevant information across multiple documents for a given search term found in a query to an LLM. Keyword searches in themselves are inefficient due to their need for exact representation. Here, NLP combines with Machine Learning to help convert text into numbers, therefore making them more amenable and easier to search. Also known as vectorisation, machine learning techniques help generate numerical embeddings of unstructured documents across multiple dimensions, which are then accessed through vector search techniques to filter data prior to making the LLM call.

Given all the above preparatory activity, there exist numerous ways through which the actual interaction with an LLM can ensue. The choice of approach depends upon the use case and the nature of questions asked. In a way, they relate to the process of providing context to an LLM for it to provide relevant and accurate results. Ranked in order of relying entirely on the LLM's knowledge to providing more context, these include:

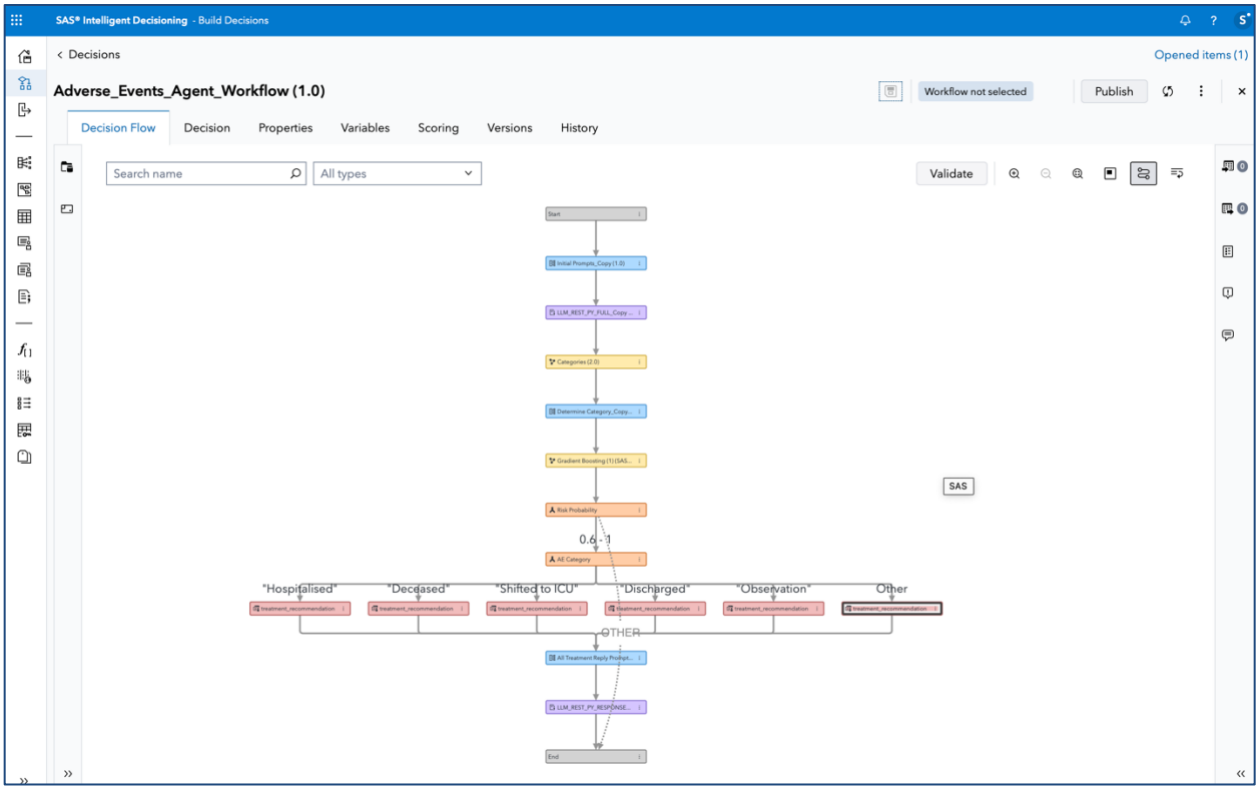
1. **Zero-shot prompting**: As the name implies, the LLM relies upon its training knowledge (in-built) to answer questions posed by the user and does not need to use external context. Examples of suitable prompts for zero-shot include asking for definitions of standard terms, or explanation of concepts. The LLM responds to the prompt

within the boundaries of its trained knowledge. Any new development that emerges after the training date is considered out of context.

2. **One-shot and few-shot prompting**: Collectively falling under the umbrella of in-context Learning, they represent an approach where a few examples of desired output are provided to the LLM, to guide its reasoning to answer the current prompt. Based on the examples, external context is provided for the LLM to use to answer the desired prompt. These are suitable for cases where the LLM needs to be nudged to provide answers in a desired format and structure (for example, extract symptoms and provide as a bulleted list).

3. **Retrieval Augmented Generation (RAG):** In this approach, the initial prompt is first used to retrieve all surrounding context which might be useful for answering the prompt and then submit them to the LLM along with specific instruction embedded in the prompt. The core difference between RAG and the above In-context learning approaches is that there exists more fuzziness around the context provided to the LLM in RAG, as it is consolidated based on a retrieval operation. The other aspect to account for is that the retrieval operation, without sufficient data preparation and filtering, can easily lead to a bloat in context, thus making it imperative that the NLP processes designed upstream are rigorously designed in order to make sure that the right level of filters are applied and only relevant and accurate context is passed to the LLM for processing.

While these are the most common patterns observed, the reader is also encouraged to refer to additional discussions (with new additions emerging at rapid frequency) and approaches, such as Chain-of-Thought reasoning and fine-tuning LLMs. The role of LLMs as the chief orchestrator of an agent-centric approach should also be examined further. Agents are required because LLMs are equipped for carrying out activities which generate text in response to written instructions. LLMs are not capable of executing actions which may be recommended in such responses. Furthermore, LLMs may sometimes need to reach out to other data sources or perform other operations (such as data manipulation) in case they do not have the answers ready within their knowledge base. For this purpose, organisations should look to leverage agent-centric approaches as much as possible to provide greater autonomy to LLMs but in a controlled manner. For example, a decision flow such as the below can assign different treatment and process-related recommendations as determined by an LLM along with other operations it uses as a basis to frame its answers.



HOW DO YOU TRUST YOUR LLM RESULTS?

A third consideration relates to the important aspect of ensuring trust in AI results. The inherent opacity and the black-box nature of LLMs raises valid concerns around their trust, reliability and controllability. While LLMs have brought AI into the forefront and have expanded the scale and complexity of tasks you can automate, the core operations at the

heart of this technology are rooted in analytical techniques which have existed for a long time, such as Natural Language Processing, Machine Learning and deep learning.

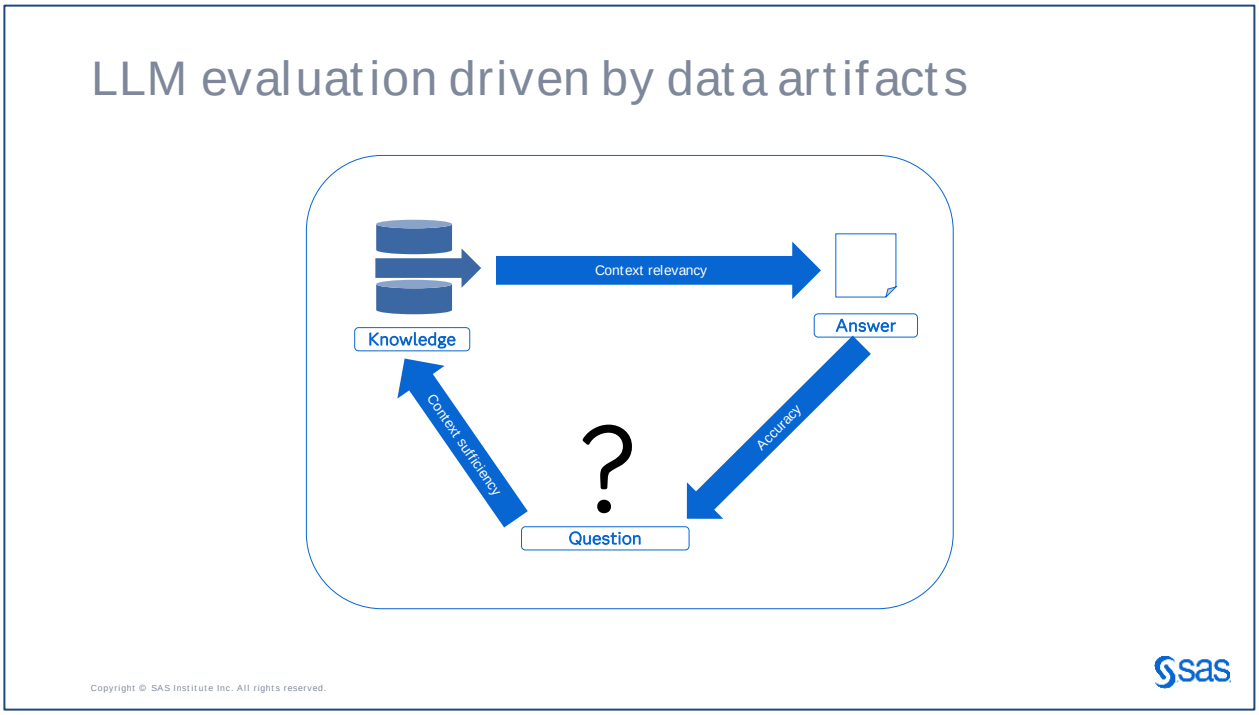
A granular understanding of the above techniques may help us consider operational, regulatory, efficiency and trust-related risks. There are two drivers behind this risk assessment,

1. There are numerous LLMs available including both commercial and open-source offerings all of which follow different standards and interfaces and might differ in focus areas.
2. When we consider how ***enterprises*** interact with LLMs, a simple question-answer paradigm might help for general questions, but for business problems, data and knowledge lies within the organization and are served by other applications and processes that directly impact the result.

All of this implies that given the multiple pieces involved, we need to identify the right context and supporting evidence to provide to an LLM and focus on a unified analytics platform which can interact with many LLMs in an efficient and consistent manner.

This also implies that organizations need to consider how to ingest data, process data and transform data into formats understood by the LLM and provide easy retrieval of such context. All these impact key metrics of accuracy, trust, and cost of solving enterprise challenges.

A suggested paradigm under which we can measure and assess trust in results would be to look at the different data elements that comprise an LLM interaction.



The three main data artifacts that are involved in an LLM interaction are:

1. **Question:** The question that is asked by the user to the LLM.
2. **Answer:** The response provided by the LLM to the user
3. **Knowledge:** The **context** (or even illustrative examples) which are provided along with the question over to the LLM for the LLM to use when crafting a response.

These above define the dimensions on which the models are evaluated. Some examples which revolve around accuracy of LLMs are provided as follows:

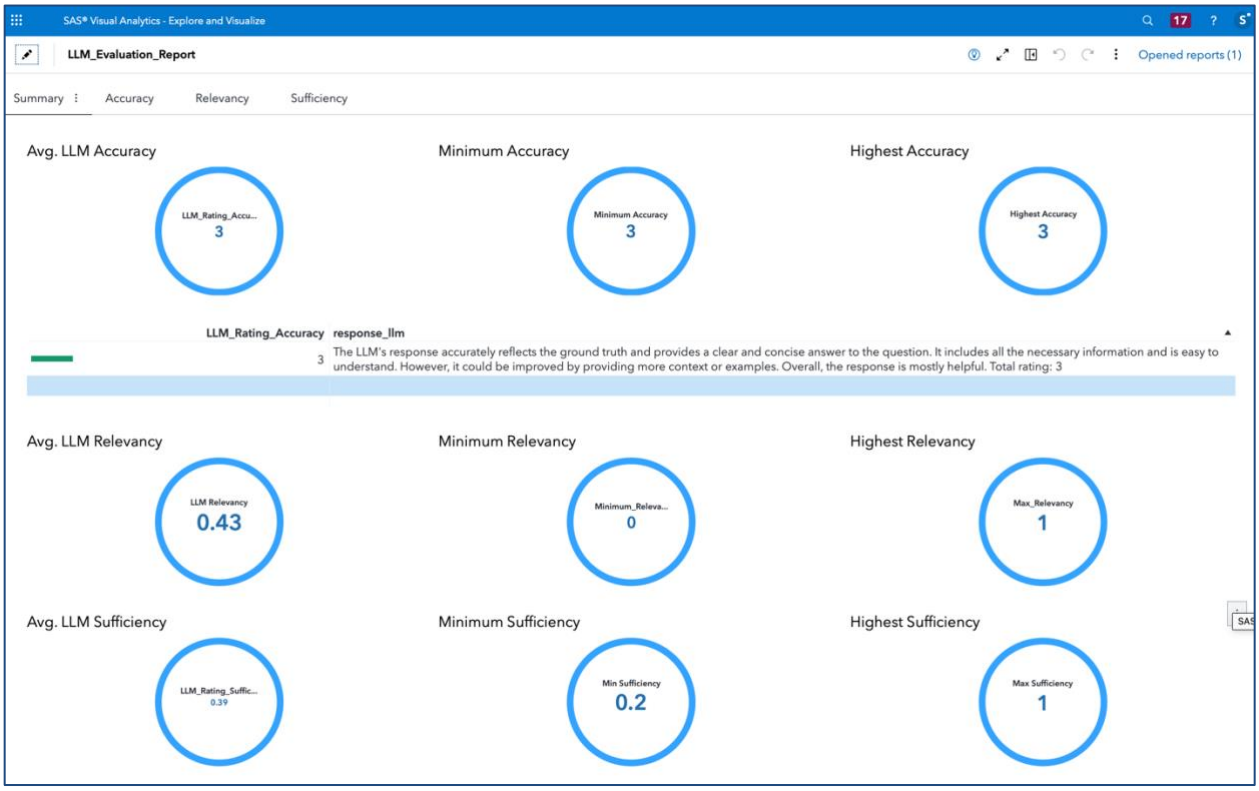
1. **Context sufficiency**: This indicates whether the context that was submitted was enough to answer the question raised by the prompt. This is driven by the context generation mechanism which can either be directed by the user or generated through a retrieval mechanism (in the case of RAG). Monitoring context sufficiency helps us understand the quality of the retrieval mechanism.

2. **Context relevancy**: This indicates if the context provided to the LLM was relevant and useful in answering the question. LLMs may sometimes not use the context provided but form answers based on their own knowledge systems, or even fabricate answers, a tendency known as hallucination. Monitoring context relevancy helps us understand the LLM's flexibility in adapting answers based on provided context.

3. **Accuracy**: This is a core metric which identifies if the response by LLM correctly answered the question posed by the user. Irrelevant and inaccurate answers are both penalised in this process. This can be measured both based on similarity (distance) metrics or based on ground truth labels, where such are made available.

These are just examples around one dimension of accuracy, but several other metrics can be designed to evaluate other aspects of LLM performance such as performance, speed, style etc.

An example evaluation dashboard, created with SAS Visual Analytics shows how these different metrics can be set up to be consistently evaluated and monitored. This helps us keep track of LLMs performance and assess whether results can be trusted sufficiently enough to implement LLMs into production processes.



CONCLUSION

Life Sciences organisations have started to seriously consider incorporating Large Language Models into various business processes to reap gains from automation and improve quality. As mentioned before, however, the world of LLMs is vast and rife with clutter, different standards and quality, and pose inherent risks with respect to how their results can be trusted. Navigating this landscape requires answering challenges individually on multiple fronts, of which we provide three which help you in this regard. Note that these considerations are influenced heavily by your target use cases, available resources and skills with respect to LLM integration and orchestration.

The authors continue to monitor developments in this area and identify places where analytics and AI is enhanced and optimised through judicious application of LLMs.

REFERENCES

1. "Safety Signals from Patient Narratives PLUS", Eid, Sherrine and Sankaran, Sundaresh, Pharma-SUG 2024, Paper SD-412, https://pharmasug.org/proceedings/2024/SD/PharmaSUG-2024-SD-412.pdf

2. “LLM – Azure OpenAI Retrieval Augmented Generation”, SAS Studio Custom Step, Sankaran, Sundaresh, 2023, https://github.com/sassoftware/sas-studio-custom-steps/tree/main/LLM%20-%20Azure%20OpenAI%20RAG

3. “LLM – Prompt Catalog”, SAS Studio Custom Step, Xin Ru Lee, 2023, https://github.com/sassoftware/sas-studio-custom-steps/tree/main/LLM%20-%20Prompt%20Catalog

4. “Evaluating Safety Compliance in Clinical Trials by Leveraging Patient Narratives and Deep Learning”, Eid, Sherrine and Sankaran, Sundaresh, Phuse 2024, Paper ET-19

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Sundaresh Sankaran
Company: SAS Institute
Address: 100 SAS Campus Dr Cary NC 27513
Work Phone: +1 919 531 4921
Email: Sundaresh.sankaran@sas.com
Website: www.linkedin.com/in/sundareshsankaran

Author Name: Sherrine Eid
Company: SAS Institute
Address: 100 SAS Campus Dr Cary NC 27513
Work Phone: +1 919 531 3991
Email: Sherrine.Eid@sas.com
Website: https://www.linkedin.com/in/sherrineeid/