Integrating Responsible AI Practices Into LLMOps.

AUTHORS

Debmalya Biswas, Dipta Chakraborty, Bhargav Kumar Mitra,
Pushpa Ramachandran M

Table Of Contents

Introduction

While Large Language Models (LLMs) have great potential to impact enterprise value chains, they have so far been mostly limited to proof-of-concept (POCs). To truly scale GenAI use cases, enterprises need to establish scalable LLM platforms with LLMOps capabilities and appropriate LLM governance.

Responsible AI will be a key aspect of LLMOps. Without the right guardrails in place, LLMs can lead to biased, misguided, and even toxic outputs.

Here, we introduce LLMOps and explore the relevant LLMOps architectural patterns. We consider the relevant dimensions of Responsible AI (such as fairness, data privacy, and explainability) and explore how organizations can achieve an integrated LLMOps capability governed by Responsible AI practices.

MLOps VS. LLMOps

In an enterprise context, MLOps provides the tooling and frameworks to scale AI/ML use cases. MLOps manages the end-to-end ML lifecycle at scale by enabling technology for the development, deployment, monitoring, and ongoing management of ML models. However, LLMOps (MLOps for LLMs) is very different from MLOps and poses many challenges that are difficult to address by legacy MLOps capabilities, such as:

Data leakage

The generative nature of LLMs means that new content is generated in real-time based on user prompts/responses. These prompts and responses can then be used to further train LLMs. However, enterprises need to ask: Who owns these conversations, and what privacy parameters need to be in place? When trained on conversations that contain sensitive data, LLMs may run the risk of leaking that data in the future.

Unstructured data

Supervised (predictive) ML primarily deals with structured data (in the form of labeled relational data, time series data, etc.) on which models are trained. In the case of LLMs, we are mostly dealing with unstructured data, documents, and multi-modal data consisting of text, images, audio, and video files.

Hallucination

Enterprises also need to verify that proper guardrails are in place to filter incorrect responses (hallucination) produced by LLMs.

Pre-trained foundational LLMs

Instead of training ML models from scratch, the most prevalent scenario is to fine-tune foundational LLMs trained on a large corpus of generic data. For teams running LLMOps, experience with traditional ML models does not necessarily translate to success fine-tuning LLMs.

Unsupervised LLM programs

Human feedback loops are an essential part of training LLMs and continuously improving the quality of LLM responses. Reinforcement learning helps to perform this improvement in a targeted fashion by leveraging a rewards strategy (e.g., Proximal Policy Optimization) to fine-tune the responses.

Responsible LLMOps

GenAI solutions are varied in scope, and we expect them to enter the enterprise landscape via different applications and platforms. Enterprise users may engage with LLM based-application directly, as in the case of ChatGPT. On the other hand, LLMs will also be embedded within SaaS products and enterprise platform (e.g., Salesforce and ServiceNow). Or enterprises might leverage foundational models fine-tuned with enterprise data for specific strategic use cases.

Recognizing the diversity of LLM architectures, this report first identifies and outlines the five most prevalent Gen AI architectural patterns: black-box LLM APIs, embedded LLM apps, fine-tuned LLMs and domain-specific LLMs, Retrieval Augmented Generation (RAG), and AI agents .

From a Responsible AI perspective, we then identify the key challenges in integrating Responsible AI dimensions (such as fairness, reliability, explainability, privacy, and security) into the outlined GenAI architectural patterns. Explainability, for example, becomes more challenging given the large corpus of (generic) training data involved in the case of LLMs. Novel privacy challenges arise given that user inputs can potentially be used as training data to fine-tune the LLMs. Hallucinations remain the key issue to be addressed given the generative nature of LLMs.

To address the above challenges, we propose a consolidated Responsible AI framework for LLM platforms, taking into consideration best practices and design patterns that integrate necessary governance and guardrails at various stages of the LLMOps pipeline.

LLMOps Architecture Patterns

Black-Box LLM APIs

This is the classic ChatGPT scenario, where enterprises gain black-box access to a LLM API/UI. Similar LLM APIs can be considered for other Natural Language Processing (NLP) core tasks, such as knowledge retrieval, summarization, auto-correct, translation, and Natural Language Generation (NLG).

Prompts are the primary interaction mechanism here and can encompass user queries and tasks. For this reason, prompt engineering is becoming a full-scale professional discipline. Prompt engineers can systematically perform trials, record their findings, and arrive at prompts that elicit the optimal contextual responses.

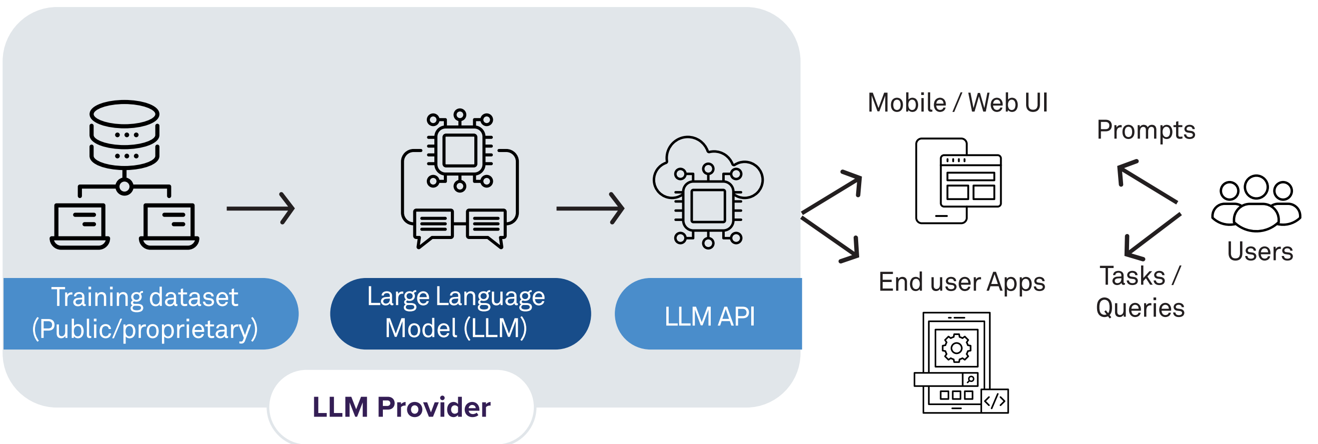


Fig: LLM APIs

Embedded LLM Apps

LLMs can be embedded within enterprise platforms such as Salesforce, SAP, and ServiceNow. They also underpin the ready-to-use enterprise apps created by LLM providers such as OpenAI.

Enterprise LLM apps have the potential to accelerate LLM adoption by providing an enterprise-ready solution. However, enterprises need to approach these applications with the same caution that should be exercised when using a third-party ML model. That means validating LLM/training data ownership, IP, and liability clauses and considering potential risks like bias and hallucination.

Data ownership is a particularly important dimension. Data is critical for supervised AI/ML systems, and especially so for LLMs, which are often trained on public datasets. The data usage rights for this AI/ML training are often not well defined and will evolve in the future as data originators seek to secure and often monetize their data ownership. For example, Reddit recently announced that it will begin charging enterprise AI/ML models to train on its massive library of user-generated content.

Data ownership issues go beyond training data. With embedded LLM apps, enterprises will also need to clarify the ownership of input data, output data, and other generated data. Further, it will be important to understand and assess how the enterprise app provider will be capturing data related to user interactions with the application.

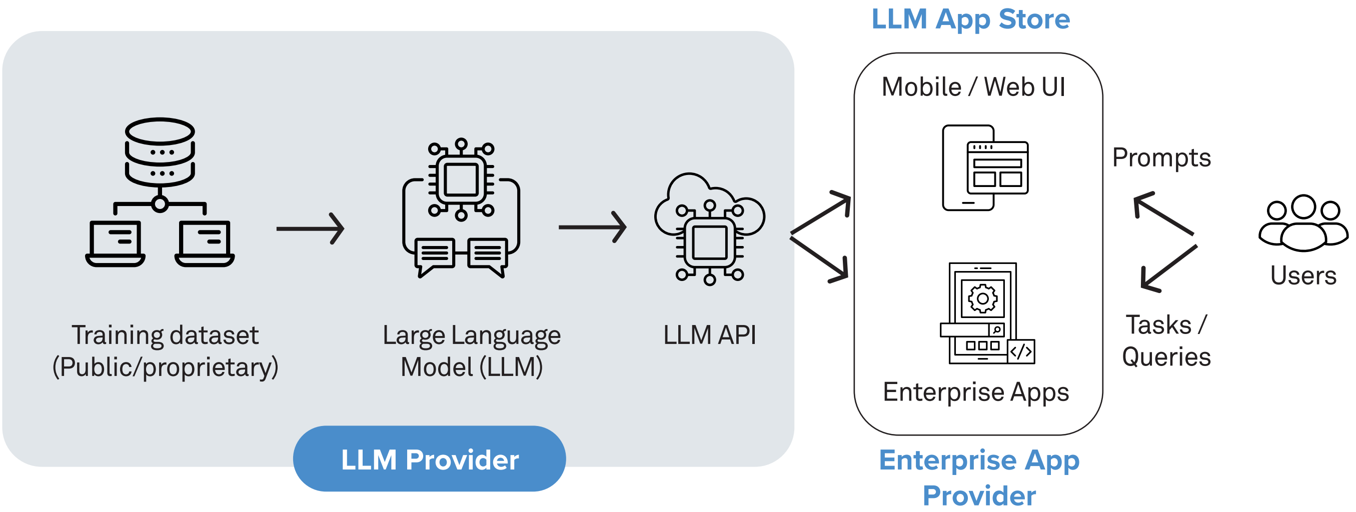


Fig: LLM apps embedded within Enterprise Apps / Platforms

Fine-tuned LLMs / Domain-specific SLMs

LLMs are generic in nature. To realize the full potential of language models, enterprises need to further train these models with enterprise-specific knowledge (data flowing through the business in the form of documents, wikis, business processes, etc.).

This contextualization is achieved in most cases by fine-tuning a Large Language Model (LLM) with enterprise data, or by creating a domain-specific Small Language Model (SLM).

Select pre-trained Large Language Model (LLM):

- OpenAI
- Google
- Meta AI
- Microsoft
- Research
- Hugging Face
- Anthropic

Specify NLP task:

- QA/Chatbots
- Extraction
- Summarization
- Auto-correct
- Translation
- Classification

Prep enterprise training data:

- Prioritize use-case(s)
- Gather training & validation data
- Annotate data

Select contextualization strategy:

- Fine-tuning
- Retrieval Augmented Generation (RAG)
- Reinforcement Learning with Human Feedback (RLHF)

Deploy enterprise model:

- Taking enterprise data into context applying selected fine-tuning strategy.

Fig: Enterprise LLM contextualization strategy

Fine-tuning re-trains a pre-trained Large Language Model (LLM) on a (smaller) set of enterprise data. On a technical level, this means amplifying and regularly updating the weights of the enterprise-specific data layers while freezing the weights of the trained neural network's base data layers. This fine-tuning yields a LLM more capable of performing enterprise-specific tasks.

On this front, open-source pre-trained LLMs present a promising opportunity for many enterprises. Meta, for example, recently open-sourced its LLaMA LLM. The Stanford Alpaca project showed that it is possible to fine-tune LLaMA for a mere $600, resulting in a model performance comparable with ChatGPT. It is becoming clear that fine-tuning a LLM does not necessarily need to be complex or expensive.

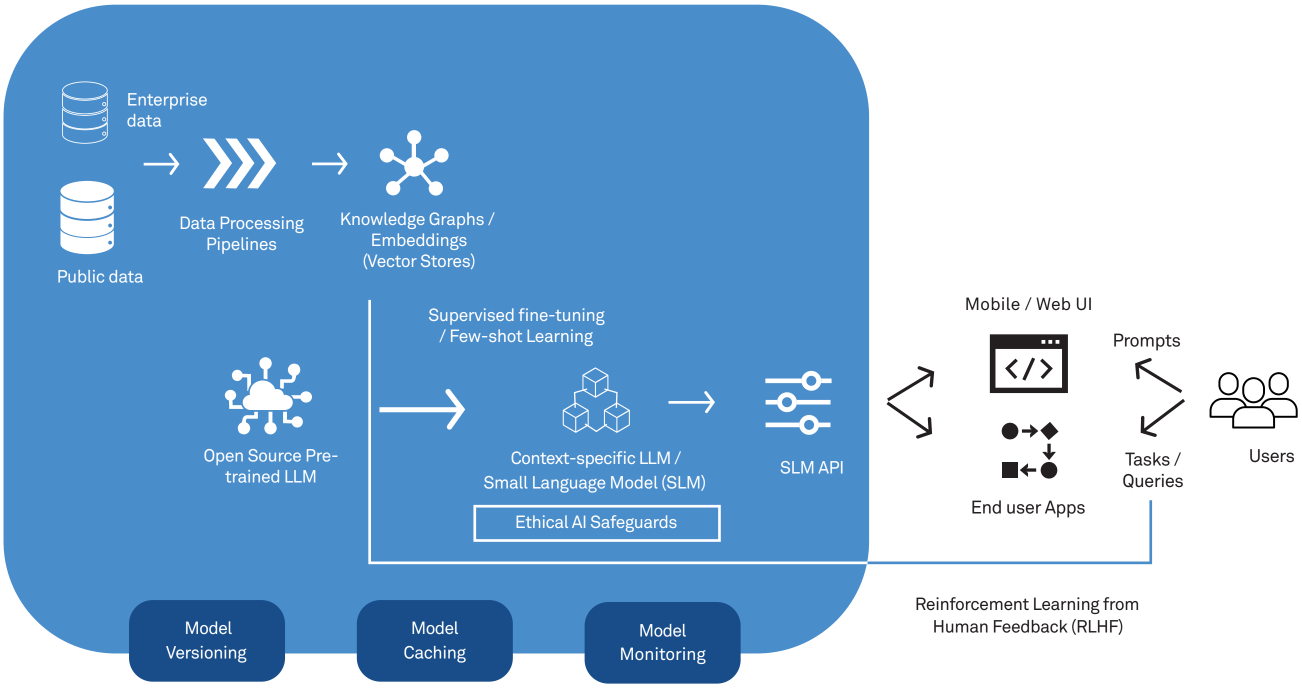


Fig: LLM Fine-tuning with Enterprise data

Retrieval-Augmented Generation (RAG)

Fine-tuning is not always feasible, as many LLMs do not accommodate fine-tuning. RAG provides a viable alternative to fine-tuning by providing additional context via prompts, grounding the retrieval/response process to a given context. This can be in the form of a set of documents that are first retrieved using an indexed document or vector search, and then provided as context with the prompts to limit the responses. Most LLM platforms now allow prompts to be rather extensive, so it is possible to embed this enterprise context as part of the prompt.

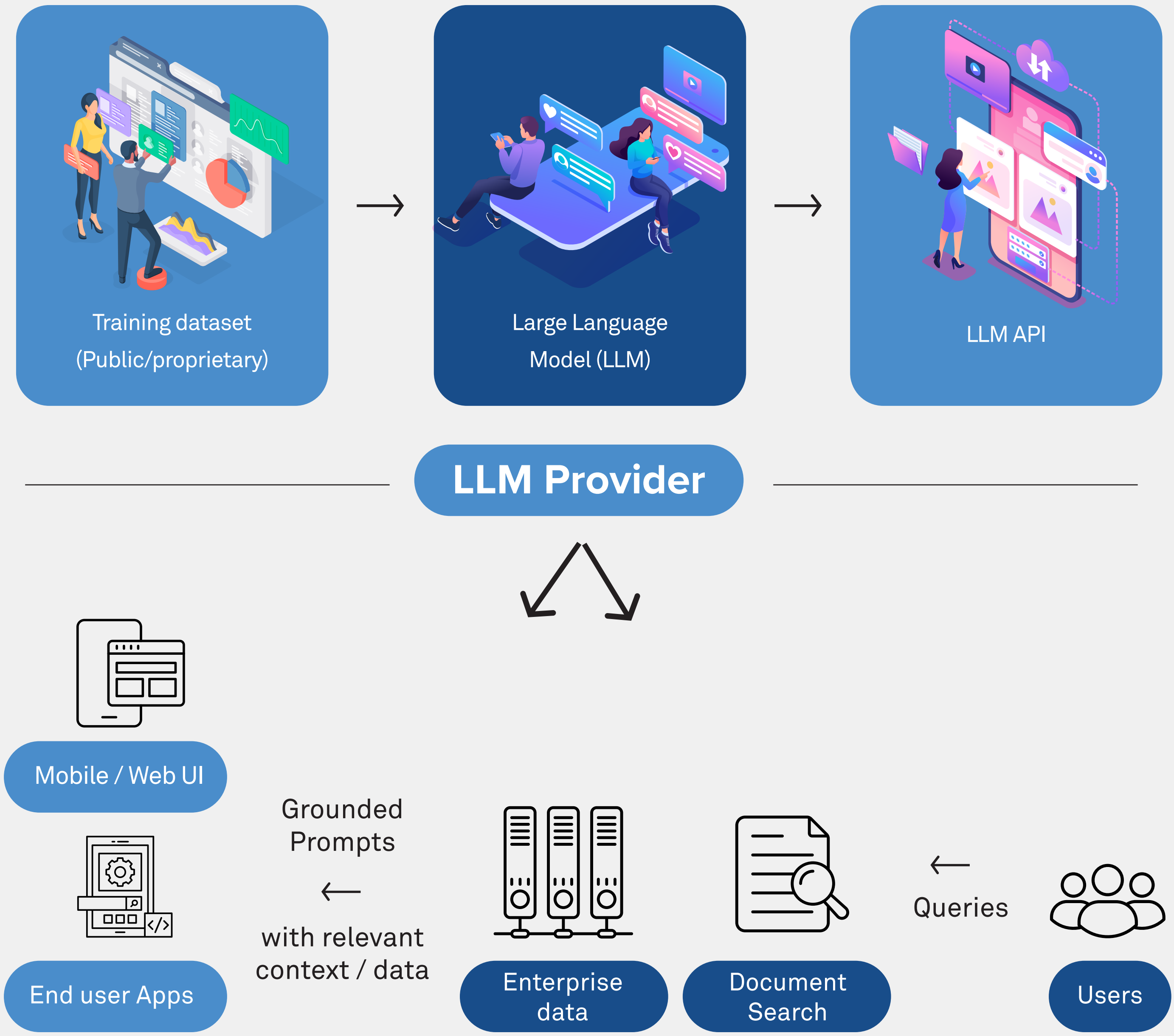
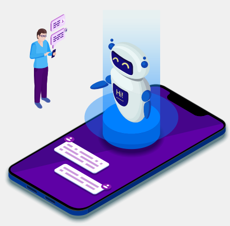


Fig: Retrieval-Augmented Generation (RAG)

A Responsible AI Framework For LLMs

The growing adoption of Generative AI, especially LLMs, has reignited the discussion around AI regulations. Regulators are seeking to ensure that AI/ML systems are responsibly trained and deployed. Unfortunately, this effort is complicated by different governmental organizations and regulatory bodies releasing their own guidelines and policies with very little agreement on the definitions of terms. Often, regulatory language is (sometimes intentionally) kept at such an abstract level that it is difficult to map to an implementation/enforcement mechanism. For example, the EU AI Act mandates best practices and restrictions based on the "risk level" of an AI application. However, quantifying the risk level of an AI application is easier said than done, as it depends on multiple parameters. Quantifying risk requires organizations to classify how the capabilities of a non-deterministic system will impact users and systems that might interact with it in the future.

Responsible LLMOps aims to ensure ethical AI deployment that avoids harmful applications, mitigates biases to promote fairness, and maintains transparency about model development and usage. Data privacy is a priority, with strict adherence to regulations to protect user information. Continuous monitoring of AI models ensures they perform accurately and ethically, while user feedback is integrated to refine and improve the models, ensuring they meet user needs and expectations. The table below summarizes the key imperatives in implementing Responsible AI for the different GenAI architectural patterns.

Responsible AI Dimensions	Factors	LLM APIs	Fine-tuned LLMs	LLMs with RAG
				
Reliability	Data Consistency	Achieve consistency through few-shot prompting or other prompt instructions.	Provide consistent and balanced training data.	Provide consistent fetched data from the vector database according to the use case and prompting.
	Bias/Fairness	Avoid training prebuilt LLMs on data that might perpetuate or amplify harmful biases.	Fine-tune LLMs with unbiased data and generate business-contextual synthetic data using LLMs to minimize the risk of biased responses.	Provide RAGs with unbiased data to reduce the chance of biased responses.

Reliability	Hallucination	Ensure that hallucinations are minimized with few-shot prompting and other prompt instructions. Because the model generates responses based on previous training data, an enterprise's ability to mitigate hallucinations is inherently limited.	Reduce hallucinations to some extent by re-training the model with curated enterprise data.	Via RAG architecture, reduce the hallucination by providing relevant data specific to use case and business domain.
	Accountability	Humans should be in the loop while training LLMs (or during the build phase) to ensure that the output of the model can be verified before deployment. Human feedback can then be leveraged for continuous improvement of the model.		
Reproducibility	Evaluation during Training	Evaluate LLM performance either through manual testing (keeping humans in the loop) or using statistical measures. There are different statistical measures available to evaluate the performance of LLMs: Perplexity, BLEU, ROGUE, etc.		
	Inference Evaluation	Measure LLM performance while handling incoming requests, using evaluations such as completed requests per minute, time to first token (TTFT), inter-token latency (ITL), and end-to-end latency.		
Explainability	Evidence	Chain of Thought (CoT) prompting can be used to ascertain the logic behind the LLM response.	Chain of Thought (CoT) prompting can be used to ascertain the logic behind the LLM response.	RAG plus CoT prompting can provide the logic to the LLM response. It can also provide the evidence link with the response.
Security	Data Guardrails	Leverage secure hyperscaler cloud platforms (e.g., Azure, AWS, GCP) to train and deploy LLMs. These platforms provide the necessary landing zones, infrastructure, and tooling to securely host both data and models.		
	Access Control	Configure context-aware access controls for the data used by LLMs as per the use case. Cloud platforms can provide a default standard protocol to access the data used in LLMs.		

Below, we expand on how each of these four dimensions of Responsible AI governance can be implemented in the LLMOps pipeline.

Data Quality (Reliability)

With respect to data quality/reliability, enterprises need to consider how various dimensions of data quality can enhance the reliability of the overall LLM ecosystem. Whether training a LLM or actively using a LLM via prompt engineering, data reliability is of utter importance. Data reliability is the foundation of trustworthy LLM outputs. The following data quality checks should be considered when seeking to use or train LLMs:

Data Consistency

To ensure accurate and precise LLM outputs, the data used for training (and particularly fine-tuning) a LLM should be relevant to the specific use case. For example, if the use case is to generate a summary of a medical prescription, the LLM should only be trained on medical prescriptions and corresponding summarizations of the prescriptions, not unrelated diagnostic information. Extra caution should be exercised when incorporating time-related data, ensuring that data is temporally relevant and refreshed at an appropriate frequency. Often, data pipelines will need to be created to automatically ingest data and feed that data to LLMs. In such scenarios, extra caution needs to be exercised while consuming running text fields, as these fields may contain a significant amount of inconsistent and incorrect data.

Bias/Fairness

In terms of model performance and reliability, controlling undesired biases in black-box LLMs is challenging. However, this can be mitigated to some extent by using fair and unbiased data for fine-tuning and/or generating business-contextual synthetic data with LLMs to reduce the likelihood of biased responses. Additionally, contextualizing the LLMs within a RAG architecture can further help manage biases.

Hallucination

When using LLM APIs or orchestrating multiple LLMs, hallucination likelihood increases due to a lack of guardrails. The right prompts can help, but only to a limited extent. To further limit hallucinations, LLMs need to be fine-tuned with curated data and/or limited to relevant and recent enterprise data.

Accountability

To make LLMs more reliable, we recommend that humans validate LLMs outputs. Involving a human ensures that if LLMs hallucinate or provide wrong response, a human can evaluate it and make it correct.

Data Quality Case Study

Working with an airport client, Wipro created a chatbot that provides real-time information on flight status and ticket availability. User can check their flight status as well as their ticket status based on their Passenger Name Record (PNR). In this context, it is extremely important to provide correct feedback to the Users.

Initially, the baseline LLM tended to provide wrong (i.e. hallucinated) flight numbers and statuses. Following a RAG implementation that automatically embedded the PNR number, passenger name, and flight number in the prompt process, the LLM was able to consistently provide accurate flight information based on natural language prompts from users.

Model Performance (Reproducibility)

With respect to model performance/reproducibility, it is important to measure the performance of the model during both the training and the inferencing phases to evaluate model performance.

Model Evaluation during Training

To ensure the performance of the model, organizations should always measure the performance of the model during the training phase before deploying it to the production. There are several metrics that organizations can use to statistically quantify model performance, including:

Perplexity

This quantifies how well a model predicts the next piece of text in a sequence. The lower the score, better the model.

BLEU (Bilingual Evaluation Understudy)

BLEU is a metric commonly used in machine translation tasks. It compares the generated output with one or more reference translations and measures the similarity between them. The higher the score, better the model.

ROUGE (Recall-Oriented Understudy for Gissing Evaluation

ROUGE is a set of metrics used to evaluate the quality of summaries. It compares the generated summary with one or more reference summaries and calculates precision, recall, and F1 score. The higher the F1 score, the better the model.

Human Evaluation

Last but not least, human evaluation or manual testing of the model is essential to check the performance of the model with different scenarios.

Model Performance during Inferencing

Monitoring model performance during live run is essential for ensuring reliability of the model. Model performance can be monitored in two ways: 1) measuring the quality of the model outcome and 2) checking model performance in handling request.

Regarding quality of model outcomes

Regular monitoring of the model outcome is essential to make the model more robust and reproducible. Evaluations like Perplexity, BLEU, ROGUE, and others can be used to measure the model performance. If there is any significant drift in model performance, the model will need to be retrained from the last verified checkpoint. In cases of drift, the model may have forgotten what it learned previously and be over-indexing toward the most recent training data.

Checking Model Performance in Handling Incoming Request

Model performance during production (i.e. handling request/handling incoming request) is the second critical aspect of model robustness and reproducibility. Metrics that can be used to evaluate model performance include completed requests per minute, time to first token (TTFT), inter-token latency (ITL), and end-to-end latency (end-to-end latency should be approximately the same as the average output length of tokens multiplied by the inter-token latency).

Model Performance Case Study

A food and beverages company aimed to refine its product development process for its chocolate candy line by leveraging customer feedback from social media. Wipro developed a solution that utilizes a Large Language Model (LLM) to analyze and summarize user responses, blending these insights with product features to guide future product enhancements. Consistently high BLEU and ROUGE scores reflected the accuracy and relevance of the generated summaries.

User Input

"What are the general audience's thoughts on the taste and texture of our new chocolate candy?"

LLM Output

"For your new chocolate candy, users are particularly impressed with its rich, creamy texture and the balance of sweetness. However, some feedback suggests that the candy could benefit from a wider variety of flavors and a more eco-friendly packaging. To enhance your product, consider introducing new flavor combinations and using sustainable packaging materials. Additionally, conducting taste tests and gathering direct feedback from customers can provide more targeted insights for product improvements."

Model Explainability

Explainability is an umbrella term for a range of tools, algorithms, and methods that accompany AI model inferences with explanations. While explainability and transparency are characterized as “non-functional” aspects of AI models, enterprises should prioritize them highly. An enterprise should be able to explain, for example, why a ML model categorized a user in a particular user segment for advertising purposes.

Chain of Thought (CoT)

is a framework that can be leveraged to understand how a LLM is solving a problem. CoT can be implemented in two ways: user prompting and automated prompting.

User Prompting

Here, a user manually prompts the model to provide the logic about how it approaches a certain problem. The LLM has the ability to understand the instruction and provide the output as per the logic provided in the prompt.

Automated Chain of Thought Prompting

Manually handcrafting CoT can be time consuming and, in the long term, is a sub-optimal solution. Automatic Chain of Thought (Auto-CoT) can be leveraged to generate the reasoning chains automatically, thus eliminating the human intervention. An Auto-CoT process can be built by combining tactics such as question clustering (clustering the questions of a given dataset) and demonstration sampling (selecting a representative question from each cluster and generating a reasoning chain). Auto-CoT works well for LLMs having approximately 100B parameters or more, but is not as accurate for small-scale LLMs.

Model Explainability Case Study

In the context of medical insurance, it is critical to ensure that LLM outputs are consistent with legally binding member agreements. Wipro created a chatbot for a medical insurance company that allows user to check the status of a claim by providing the claim ID. The chatbot returns not only the status of the claim, but also a verifiable reference to the relevant policy language, which ensures explainability.

User Input

“What is the status of claim ID 123456789?”

LLM Output

“Upon checking, we can confirm that your claim is approved and 80% of the claim amount is finalized for payment in accordance to the condition mentioned in the policy terms section 1.5.”

Data Privacy (Security)

With respect to data privacy, enterprises need to consider the privacy aspects of both enterprise data used to fine-tune the LLMs and enterprise data provided as context (RAGs). In addition, one novel privacy aspect of GenAI is the need to consider the privacy risks of data (prompts) provided voluntarily by the end users, which can potentially be used as training data to re-train/fine-tune LLMs.

Cloud providers and hyperscalers providing LLMs and enabling their fine-tuning on enterprise data can provide the necessary setup/landing zone for data privacy and access controls access to the data for specific use cases.

Conclusion

Gen AI is a disruptive technology, and we are seeing it evolve faster than any technology innovation curve we have experienced before. As such, it is critical for organizations to understand the key architectural patterns entailed in scaling GenAI proofs-of-concept into production. A mature LLMOps strategy will be critical to both accelerating innovation and mitigating risk. Large enterprises will increasingly need to manage the mix of proprietary, open-source, and fine-tuned LLMs. Across all GenAI architectures, LLMOps should ensure that LLMs are being used efficiently and responsibly.

We have proposed a LLMOps framework that optimizes tooling and frameworks to deliver responsible, governed LLM capabilities. We believe that this framework will accelerate LLM adoption and enable enterprises to scale GenAI use cases in a responsible fashion. Responsible LLMOps effectively future-proofs GenAI investments and ensures that LLM platforms can cope with a GenAI landscape that will bring new LLMs and new training/deployment architectures at a rapid pace.

Ambitions Realized.

Wipro Limited
Doddakannelli
Sarjapur Road
Bengaluru – 560 035
India

Tel: +91 (80) 2844 0011
Fax: +91 (80) 2844 0256
wipro.com

Wipro Limited (NYSE: WIT, BSE: 507685, NSE: WIPRO) is a leading technology services and consulting company focused on building innovative solutions that address clients' most complex digital transformation needs.

Leveraging our holistic portfolio of capabilities in consulting, design, engineering, and operations, we help clients realize their boldest ambitions and build future-ready, sustainable businesses. With over 230,000 employees and business partners across 65 countries, we deliver on the promise of helping our clients, colleagues, and communities thrive in an ever-changing world.

For additional information, visit us at **www.wipro.com**

GEO/AGENCY CODE/VALIDITY