A Practical and Pragmatic Approach for Responsible AI Usage and AI Governance for GenAI for Small and Medium Businesses, Communities and Individuals

Koustav Bhar

Abstract:
As GenAI becomes common-place, usage of Artificial Intelligence (AI) is becoming increasingly crucial for small and medium businesses, communities, and individuals. This paper proposes a practical, cost efficient, and privacy-conscious framework for evaluating application systems built on Large Language Models (LLMs). Traditional machine learning models are typically assessed using standardized quantitative metrics such as accuracy, precision, recall, F1-score, RMSE, and R^2, where outputs are structured and easily comparable to labeled ground truth. In contrast, LLM based systems generate open-ended, context-sensitive responses, making their evaluation significantly more complex. Measuring performance in such systems must go beyond correctness alone and include dimensions such as factual grounding, consistency, bias control, safety behavior, resistance to malicious prompts, and ethical alignment. The growing enterprise adoption of GenAI solutions creates an urgent need for an evaluation methodology that is reliable, explainable, and economically sustainable.

The framework described in this work introduces a structured evaluation approach designed specifically for LLM-powered applications that perform extraction, reasoning, and question answering over documents. Rather than depending on complex and expensive multi-model evaluation architectures — especially those that rely on using another LLM as an automated judge — this method emphasizes controlled testing, deterministic prompt design, and ground-truth-based validation. This avoids the governance and reliability concerns associated with "who evaluates the evaluator," while also significantly reducing operational costs and architectural complexity.

The methodology begins with cross-domain document preparation, privacy sanitization, and detailed manual analysis to identify verifiable data points. From these, validated ground truth datasets and repeatable prompts are created. Documents are then segmented using an optimized chunking strategy with contextual overlap and embedded using an enterprise-approved embedding model. These embeddings are stored in a vector database to enable semantic retrieval. At runtime, user queries are matched to the most relevant content segments through similarity search, and only a small number of top-ranked chunks are supplied to the LLM along with a strict system prompt. This constrained-context design improves response relevance, reduces hallucinations, and controls token usage and latency.

Evaluation is performed by systematically comparing model outputs with predefined ground truth answers to measure accuracy and detect drift. Additional monitoring layers analyze user interactions and model responses to identify bias indicators, malicious or unethical intent, and appropriate refusal behavior. These checks are implemented through rules, controlled prompts, and scoring logic rather than secondary judging models. All queries, retrieved contexts, prompts, responses, and evaluation results are logged to support traceability, audit readiness, and stake holder reporting on performance and responsible usage. Overall, the proposed framework demonstrates that robust LLM evaluation can be achieved through a transparent, lightweight, and enterprise-aligned architecture. It balances accuracy, safety, privacy, and cost, while remaining extensible for future enhancements such as newer models, improved embeddings, refined prompts, and multi-layer evaluation strategies. By addressing these critical components, organizations and individuals can navigate the complexities of AI implementation while ensuring ethical and accountable AI practices.

Index Terms: LLM Bias and Safety Evaluation, GenAI Evaluation Framework, LLM Accuracy Evaluation, Malicious Intent Detection, Responsible AI Usage Monitoring.

I. INTRODUCTION

In traditional machine learning systems, model performance is typically evaluated using well-defined and widely accepted metrics such as Accuracy, Precision, Recall, F1-score, RMSE, and R^2. The choice of metric depends on the task type — for example, classification or regression — and the evaluation process is generally straightforward because the outputs are structured and the expected results are clearly labeled. These metrics make it relatively easy to quantify model effectiveness and compare different models objectively.

However, evaluating Large Language Model (LLM)– based systems is far more complex. Unlike conventional ML models, LLMs generate open-ended, context dependent responses rather than fixed categorical or numerical outputs. Their behavior can vary with prompt phrasing, context length, and retrieval inputs. As a result, standard ML metrics alone are insufficient for assessing the performance of applications that rely on LLMs for text generation, information extraction, reasoning, or decision support. The evaluation challenge shifts from simple prediction correctness to response quality, factual grounding, consistency, safety, and alignment. In one of the platforms I was required to build, there was a need to continuously measure and monitor multiple dimensions of system performance. This included answer accuracy, model drift over time, bias detection, and the handling of malicious or unethical user intents. Designing an evaluation process that could reliably measure these factors proved to be a non-trivial task. Each dimension required its own methodology, test design, and validation criteria. Although modern LLMs include built-in safeguards and moderation layers, these default protections are not sufficient for real world enterprise deployments with diverse and evolving use cases. Practical implementations demand additional, system-level evaluation and monitoring mechanisms tailored to the domain and risk profile. This is where a simple, structured, and practical evaluation approach becomes essential — one that enables ongoing tracking of accuracy, drift, bias, malicious intent handling, and ethical compliance in GenAI solutions without excessive complexity or cost.

II. METHODOLOGY AND IMPLEMENTATION

A. LLM Accuracy

For measuring LLM drift, a very simple and novel approach is suggested where we prepare some documents having elaborate data. These will have very specific details as well. These documents need to be fed to the system as a first step. Following this, some prompts need to be designed which will fetch the information from the documents. Here there can be multiple steps and the system can be customized to fetch data from complex data formats within a document. As the next step, since the requirements are to check for accuracy of the system designed, the ground truth would be needed. The ground truth needs to be extracted into a data source, which can be a database or a data file as needed. Then the prompts are executed against the documents and LLM responses are checked against the ground truths. The point to be noted here is that we are not using another LLM as a judge to evaluate the system which in itself had the LLM integration as the core component.

B. LLM Bias

For LLM bias measurement, the interactions that all users are having with the system is evaluated. In this case as a first step, the system reads the user's interaction which is recorded and stored in the system for governance and responsible usage tracking. These interactions are

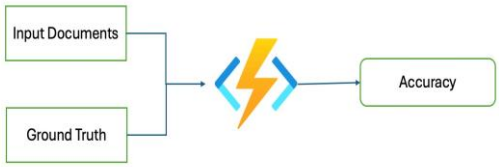

Fig. 1. Architecture Diagram - LLM Accuracy [1]

then evaluated to check for the occurrences of restricted words and phrases which would indicate that the system is displaying bias to certain gender, group, caste, creed, religion, colour, etc. if it is found that the system responds in certain ways, then further evaluation needs to e carried out against the prompts that make the LLM respond in such manner by creating a report and sharing with the intended shareholders [3][4].

C. Unethical and Malicious Content

In this case, it is checked if the user intended to ask something that is malicious and unethical for example, something that is sexually explicit or self-damaging and can cause harm to the user himself or herself or anyone else. Here the response of the LLM is also checked but that is to check if it refused to answer such a question by the user. The user is flagged in the report and it is sent to the management and stakeholders. This happens for every interaction the user intends to have with the system in question. It is again to be noted here that there is no additional LLM used as a judge where the question may arise asking how to police the police. It is a simple and practical implementation for checking any unethical and malicious content being supplied to the LLM and its response to such queries. It helps the organization to maintain the safety and security of its clients and employees equally [5][6].

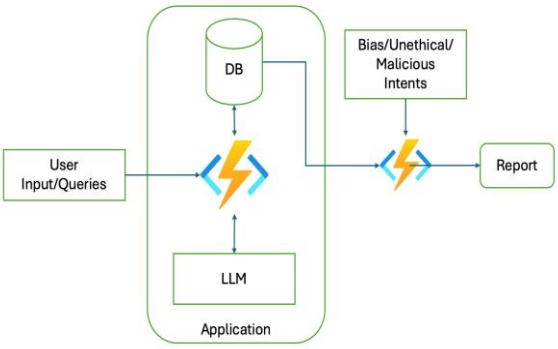

Fig. 2. Architecture Diagram - LLM Bias, Responsible Usage. [1]

III. DATA PREPARATION

The primary requirement of this solution was to enable comprehensive comparison across diverse types of data, regardless of the industry or domain from which the data originated. Rather than being limited to a single sector, the solution was designed to handle and analyze data spanning multiple domains, including but not limited to Insurance, Finance, Reinsurance, Loans, and Project Management. This cross-domain capability was a critical requirement, as it ensured that the solution could be applied in a wide range of business contexts without the need for domain-specific customization. To achieve this objective, a broad and representative collection of documents was curated as input for the solution. These inputs included publicly available documents relevant to the identified domains, such as regulatory guidelines, industry reports, policy documents, and best-practice frameworks. In addition to publicly accessible sources, proprietary documents that were restricted within the organization were also incorporated. These internal documents provided valuable real-world context and helped ensure that the solution could effectively handle enterprise-specific data structures, terminology, and use cases. All selected documents underwent a thorough review process. Each document was carefully studied to understand its structure, content, and domain-specific nuances. Based on this analysis, the information was generalized where necessary to enable meaningful comparison across domains while still preserving the core intent and relevance of the data. This generalization step was essential to ensure consistency and interoperability among documents originating from different industries. Furthermore, strict data privacy and compliance measures were followed throughout the process. Any Personally Identifiable Information (PII) identified within the documents was systematically detected and redacted. This redaction ensured that the final document set adhered to data protection standards and organizational policies. The outcome was a sanitized, standardized, and secure collection of documents suitable for accurate, cross-domain data comparison.

IV. PROMPT DESIGN AND ENGINEERING

A thorough understanding of the documents and the underlying data is a critical prerequisite for this stage of the process. Before any meaningful extraction or analysis can take place, it is necessary to clearly comprehend the structure, context, and intent of the content. This foundational understanding enables the formulation of relevant and well-defined questions that can be systematically asked of each document. Without this preparatory step, the downstream outputs risk being inconsistent, incomplete, or unreliable. To support this

objective, each document was reviewed with great care and attention to detail. Rather than relying on superficial reading, a meticulous examination was carried out to identify specific, high value data points embedded within the text. These data points were selected based on their clarity, relevance, and potential to yield precise and verifiable answers. The goal was to focus on information that could be consistently located and validated across documents, thereby improving both the accuracy and the reliability of the extracted responses. Following the identification of these target data points, the next step involved designing carefully structured prompts. These prompts were engineered to guide the model toward producing stable, repeatable answers rather than variable or interpretive ones. Precision in prompt wording was treated as essential, since even small differences in phrasing can significantly influence model outputs. The prompts were therefore tested and refined to ensure that they consistently produced the intended response format and content across multiple runs. An important factor in this workflow is the specific large language model (LLM) used to generate responses. Different models exhibit different behavioral patterns, strengths, and limitations. As a result, prompt design cannot be separated from model selection. Effective prompt engineering requires a solid understanding of how the chosen LLM interprets instructions, handles ambiguity, and structures its answers. Familiarity with the model's tendencies makes it possible to craft prompts that reliably extract the required information while minimizing noise or hallucinated content. It is also important to recognize that this setup is not universally transferable without adjustment. If any major component changes — such as the source documents, the prompt templates, the target data fields, or the LLM itself — the prompts will likely need to be redesigned and revalidated. Prompt engineering is therefore not a one-time task but an iterative and adaptive process. Future studies or system updates should plan for prompt re-engineering as a necessary step when modifying inputs or tools. This phase of the work was handled with particular rigor. Prompts were constructed using a two part structure: a system-level instruction combined with the specific user query. The system prompt established global rules for tone, format, and response constraints, ensuring that every answer followed a consistent structure. The query portion then focused on the specific information to be extracted. This layered approach helped standardize outputs across different questions and documents, resulting in responses that were uniform, comparable, and easier to validate and process downstream.

V. DOCUMENT CHUNKING

Sending entire documents directly to a large language model can be expensive and inefficient, and it often introduces practical constraints such as token limits and increased processing time. Large inputs not only raise operational costs but can also lead to incomplete processing if they exceed model limits. To address these challenges, it is necessary to divide documents into smaller, manageable segments before passing them to the model. To achieve this, a straightforward chunking strategy was adopted. Each document was split into units roughly equivalent to one page or about 1,000 characters per chunk. In addition, an overlap of approximately 200 characters was maintained between consecutive chunks. This overlap plays an important role in preserving contextual continuity, ensuring that key information spanning chunk boundaries is not lost. Without overlap, relevant details located at the edges of segments could be missed or misinterpreted during later processing steps. This chunking approach also supports more accurate semantic mapping when the text is embedded and stored in a vector database. In this case, the processed chunks were indexed within an AI Search vector index. Smaller, context-preserving chunks improve the quality of embeddings and make similarity search more precise, since each vector represents a focused portion of content rather than an overly broad document. As a result, retrieval becomes more relevant and targeted when queries are executed. Multiple experimental runs were conducted to refine this chunking logic in combination with prompt design, model behavior, and query processing workflows. Different chunk sizes and overlap values were tested to balance context retention with efficiency. Through these trials, the selected configuration proved to be the most effective for this particular solution. Overall, this step was crucial in ensuring that the system remained scalable and performant. By controlling input size and optimizing chunk structure, the solution avoided unnecessary resource consumption and prevented performance bottlenecks, resulting in a faster and more cost-effective processing pipeline.

VI. DATA EMBEDDING

Selecting an appropriate embedding model was another important step in the overall solution design. The choice was guided not only by technical suitability for the use case, but also by enterprise governance and

approval requirements. It was necessary to ensure that the selected model met internal compliance, security, and usage standards. Based on these criteria, the OpenAI ada-003 embedding model was chosen, as it was both technically well-suited for semantic representation tasks and formally approved for enterprise use. This approval status significantly influenced the final decision, since it allowed the solution to move forward without additional regulatory or procurement delays. After the document chunking process was completed, each chunk was processed through the selected embedding model to generate high-dimensional vector representations. These vectors capture the semantic meaning of the text segments rather than just their keywords, enabling more accurate similarity comparisons and contextual retrieval later in the workflow. Each generated vector was then stored in the AI Search index, which functions as the vector database for this system. By embedding and indexing the chunks in this way, the solution enables efficient semantic search and retrieval. Queries can be matched against vector representations to quickly identify the most relevant content segments, improving both accuracy and response speed in downstream applications.

VII. SEMANTIC SEARCH

At this stage, the core groundwork for the system had been completed. The source documents had been thoroughly analyzed, resulting in a strong conceptual and structural understanding of the data. Key data points were identified, a suitable chunking and embedding strategy was defined, and carefully engineered prompts were developed to guide model behavior. All document chunks were successfully embedded and stored in the vector database, making the knowledge base ready for semantic retrieval and query-driven processing. With this foundation in place, the operational workflow could be executed end to end. When a user submits a query, the system first converts the query into an embedding and performs a similarity search against the vector database. From this search, the top three most semantically similar chunks are retrieved. These selected chunks are then packaged together with the user's query and the predefined system prompt and sent to the LLM for answer generation. The decision to limit retrieval to the top three matching chunks was deliberate. The primary objective was to keep the system lightweight, fast, and cost-efficient while still providing enough contextual grounding for accurate responses. Retrieving too many chunks would increase token usage, slow response time, and potentially introduce irrelevant context. In contrast, a focused set of highly relevant chunks gives the LLM sufficient grounding information without overwhelming it. This constrained-context approach also helps reduce hallucinations by anchoring the model's response to specific, retrieved source material rather than broad prior knowledge. The prompts were explicitly designed to instruct the model to rely only on the supplied context and not on its external or pretraining knowledge. By enforcing this boundary through prompt design and retrieval limits, the system produces more controlled, verifiable, and context-driven answers while maintaining strong performance and responsiveness.

VIII. LLM COMPLETION AND EVALUATION

This phase represented the final execution stage of the solution workflow, where all previously designed components were brought together into a functioning query-and response pipeline. At this point, the LLM was invoked using three key inputs: the top three semantically similar chunks retrieved from the vector index, the predefined system prompt, and the specific user query describing the information being requested. This structured input assembly ensured that the model received focused, contextually relevant material along with clear behavioral instructions before generating its response. The OpenAI GPT-4o model was selected as the preferred LLM for this stage. The choice was based on multiple factors, including cost efficiency, response quality, latency, and overall accuracy. However, technical performance alone was not the only consideration. Enterprise constraints also played a decisive role, particularly model approval status, data privacy requirements, and regional availability rules. Only models that satisfied internal governance and compliance standards could be used in production. GPT4o met these operational and policy requirements while also delivering strong performance, making it the most suitable option for deployment. Once the model generated an answer, the output was systematically evaluated against a predefined ground truth for the given query. The ground truth represented the validated, expected answer derived earlier through manual document review and controlled prompt testing. Comparing model responses against this benchmark allowed objective measurement of accuracy and consistency. The same evaluation flow was applied across all defined metrics, ensuring uniform assessment methodology and comparable results across different test cases. It is important to note that not all evaluation categories depended on document-based context. In particular, assessments

related to malicious intent and bias detection did not rely on the document corpus as a prerequisite input. Instead, in these cases, the prompts themselves served as the primary test data. Carefully constructed prompts were used to probe model behavior under sensitive or adversarial scenarios. The evaluation criteria were adjusted accordingly to measure whether the system responded safely, neutrally, and in alignment with responsible AI guidelines rather than factual document accuracy. All interactions within this pipeline — including retrieved chunks, prompts, model responses, evaluation results, and comparison outcomes — were logged and stored in a database. This comprehensive interaction logging enabled traceability and auditability across the system's operation. The stored records were later used to generate consolidated reports for stakeholders. These reports provided visibility into system accuracy, performance trends, response quality, user interaction patterns, and indicators of responsible or potentially risky usage. By capturing and analyzing this data, stakeholders could make informed decisions about system reliability, governance compliance, and future optimization opportunities.

IX. BACKGROUND AND RELATED WORK

The final system design was developed after reviewing several existing evaluation frameworks that rely on complex architectures, particularly those that use an LLM itself as a judge to assess outputs. While such approaches appear sophisticated, they introduce a fundamental challenge: verifying the reliability of the judging model. If an LLM is responsible for evaluating another LLM's responses, it becomes difficult to independently confirm whether the evaluation is accurate and unbiased. This raises an important governance question — who validates the judge — and creates uncertainty around the trustworthiness of the evaluation results. Another major concern with these multi-layered evaluation systems is cost. Frameworks that depend on multiple model calls, cross-model comparisons, and layered scoring mechanisms can become extremely resource-intensive. In many cases, the operational cost of running the evaluation framework may exceed the cost of running the primary LLM-powered application itself. For most organizations, this imbalance acts as a significant deterrent, making such evaluation approaches impractical for regular or large-scale use. These solutions also tend to rely on multiple LLMs, often sourced from different providers and deployed across different geographic regions. This creates additional complications related to data governance, regulatory compliance, and privacy protection. When data is transmitted to several external models across jurisdictions, it increases the risk surface and complicates adherence to enterprise data handling policies. In contrast, this risk and cost profile highlights a broader limitation seen in many LLM-based offerings that depend on distributed, multi-model evaluation pipelines.

X. CONCLUSION AND FUTURE WORK

The system described above offers a privacy-preserving and straightforward alternative that is both practical to implement and easy to understand and measure. Its design emphasizes transparency and traceability, ensuring that the evaluation logic and outputs can be clearly interpreted and verified. Because the methodology is grounded in controlled prompts, validated ground truth, and measurable retrieval steps, the results it produces are dependable and reproducible. This structured approach significantly reduces ambiguity and minimizes the likelihood of incorrect or unverifiable evaluations. From a cost perspective, implementing this framework adds only a very small overhead to the primary LLM-based software solution. It avoids the heavy operational and computational expenses associated with complex, multi-model evaluation architectures, making it financially viable for enterprise adoption and continuous use. The lightweight nature of the system ensures that organizations can maintain evaluation rigor without introducing disproportionate infrastructure or model invocation costs. Looking ahead, the system can be progressively enhanced by adopting newer LLMs and more advanced embedding models as they become available and approved. Such upgrades are expected to further reduce operational costs while simultaneously improving response quality, retrieval accuracy, and overall performance. Prompt templates will also need periodic revision to keep pace with evolving model behavior and emerging best practices. In particular, prompts related to bias detection, malicious intent identification, and unethical usage monitoring should be continuously refined as new risk patterns and evaluation techniques emerge. There is also meaningful opportunity to expand the framework's coverage and strengthen its evaluation logic. Future iterations could incorporate broader scenario testing and deeper validation layers. The evaluation approach itself could be extended into a more sophisticated, multi-layered architecture. These potential enhancements, however, are

substantial enough to warrant separate, dedicated discussion and will be explored in future work.

REFERENCES:

[1] Microsoft Azure Functions image, 2026, Retrieved from [https://www.google.com/url?sa=t&source=web&rct=j&url=https]

[2] Koki Wataoka, Tsubasa Takahashi, Ryokan Ri. 2024. Self-Preference Bias in LLM-as-a-Judge. Retrieved from [https://doi.org/10.48550/arXiv.2410.21819].

[3] Lin, L., Wang, L., Guo, J., & Wong, K. (2025). Investigating Bias in LLM-Based Bias Detection: Disparities between LLMs and Human Perception. Retrieved from [https://aclanthology.org/2025.coling-main.709/].

[4] Oketunji, A. F., Anas, M., Saina, D. (2023). Large Language Model (LLM) Bias Index – LLMBI. Retrieved from [https://doi.org/10.5281/zenodo.10441700, https://doi.org/10.13140/RG.2.2.13670.80966].

[5] Banerjee, S., Layek, S., Hazra, R., & Mukherjee, A. (2025). How (Un)ethical Are Instruction-Centric Responses of LLMs? Unveiling the Vulnerabilities of Safety Guardrails to Harmful Queries. Retrieved from [https://doi.org/10.1609/icwsm.v19i1.35811].

[6] Meng Jiang, Wenjie Wang, Chongming Gao, Shaofeng Hu, Kaishen Ou, Hui Lin, Fuli Feng. (2025). An LLM-based Behavior Modeling Framework for Malicious User Detection. Retrieved from [https://doi.org/10.1145/3746252.3761520].