Introduction to RAG Validation

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5019627

AI is a Vibing Machine

Designed to be generative – to fill in blanks, finish thoughts, spin up possibilities

- Very useful for use cases with low-risk downside
- But, we want to use them for high-stake cases where mistakes can cause real harm (e.g., Finance, Medicine)

Examples: GenAI for Complaint Detection

Statement

"I really appreciate the time your team takes to ensure everything is processed correctly. I noticed my refund request is still pending after two weeks, which is longer than I expected. I'm sure there's a good reason, but could you confirm if everything looks good on my end?"

LLM Evaluation

Based on the message provided, this does not appear to be a complaint. The customer is expressing appreciation for the team's efforts, and is politely inquiring about the status of their refund request, which has been pending for longer than expected. The tone is courteous and the customer acknowledges there may be a good reason for the delay. This seems to be a request for information and clarification, rather than a complaint.

Examples: GenAI for Complaint Detection

Statement

"I really appreciate the time your team takes to ensure everything is processed correctly. I noticed my refund request is still pending after two weeks, which is longer than I expected. I'm sure there's a good reason, but could you confirm if everything looks good on my end?"

LLM Evaluation

This message is a complaint because, although the customer is polite and expresses appreciation for the team's efforts, they are still expressing dissatisfaction with the service they received. Specifically, they mention that their refund request has been pending for longer than they expected (two weeks), which implies that they are unhappy with the delay.

The customer's tone is courteous and they give the benefit of the doubt by suggesting that "there's a good reason" for the delay, but the underlying issue is still a problem with the service. They are also seeking confirmation that everything is in order, which suggests that they are seeking a resolution to the issue.

The complaint is implicit, rather than explicit, but it is still a complaint because the customer is bringing a problem to the attention of the company and seeking a solution.

Examples: GenAI for Complaint Detection

System Prompt - Before

Will you please identify if the following message from the customer is a complaint or not? Only answer yes or no.

System Prompt - After

Will you please identify if the following message from the customer is a complaint or not? It is possible that the customer can be polite and still be complaining. only answer yes or no.

Examples: GenAI for Complaint Detection

Statement

"I really appreciate the time your team takes to ensure everything is processed correctly. I noticed my refund request is still pending after two weeks, which is longer than I expected. I'm sure there's a good reason, but could you confirm if everything looks good on my end? f"

LLM Evaluation

Although the customer starts with a positive statement, expressing appreciation for the team's efforts, the main concern they're raising is that their refund request has been pending for two weeks, which is longer than they expected. This indicates a issue or dissatisfaction with the processing time, which is a key aspect of a complaint.

The customer is politely asking for confirmation on the status, but the underlying sentiment is one of concern and potential frustration with the delay. The phrase "which is longer than I expected" specifically highlights their disappointment with the timeframe, which further supports the classification of this message as a complaint.

Comprehensive Validation of an LLM is Impossible

No control on training data

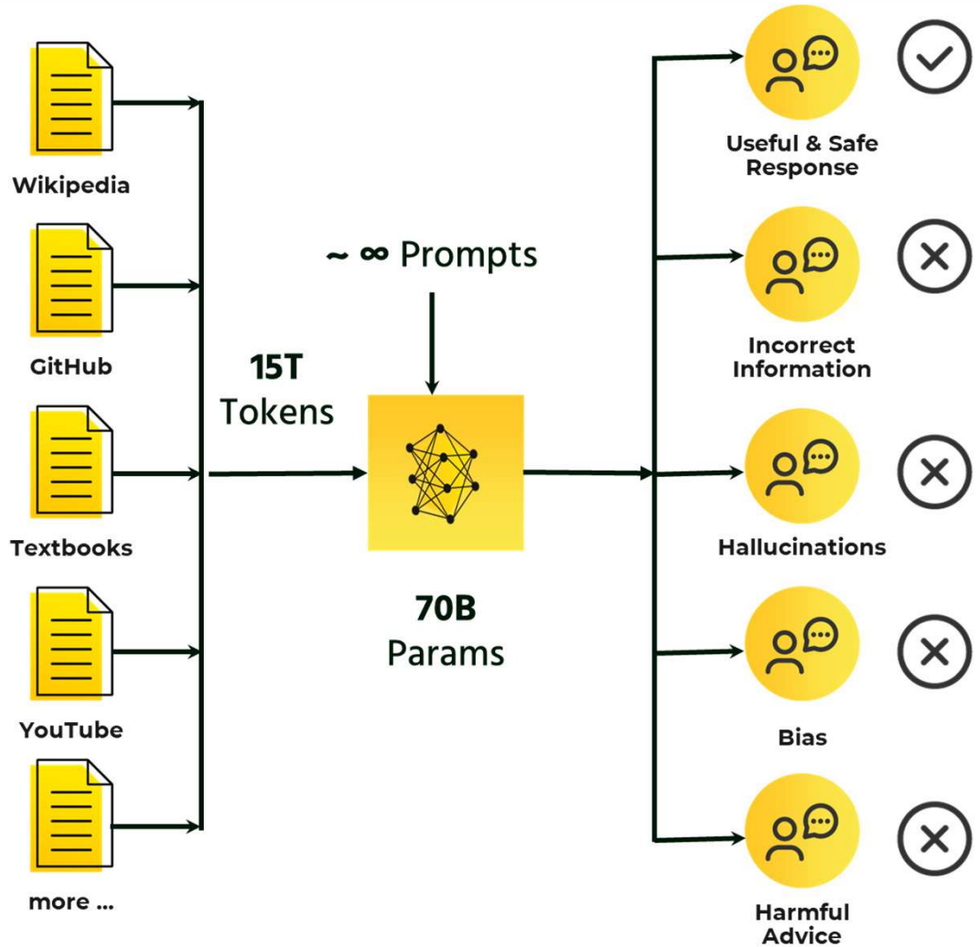
Unlike predictive AI where training data is very well curated, training data for LLMs is comprised of data from varied sources.

No control on inputs: infinite prompting possibilities

Unlike predictive AI where input variables are well selected and bounded, a generic LLM chat bot is open ended and unbounded.

Wide range of outputs

Unbounded outputs are difficult to evaluate comprehensively and control even with guardrails



Comprehensive Validation of RAG is possible

RAG is Bounded

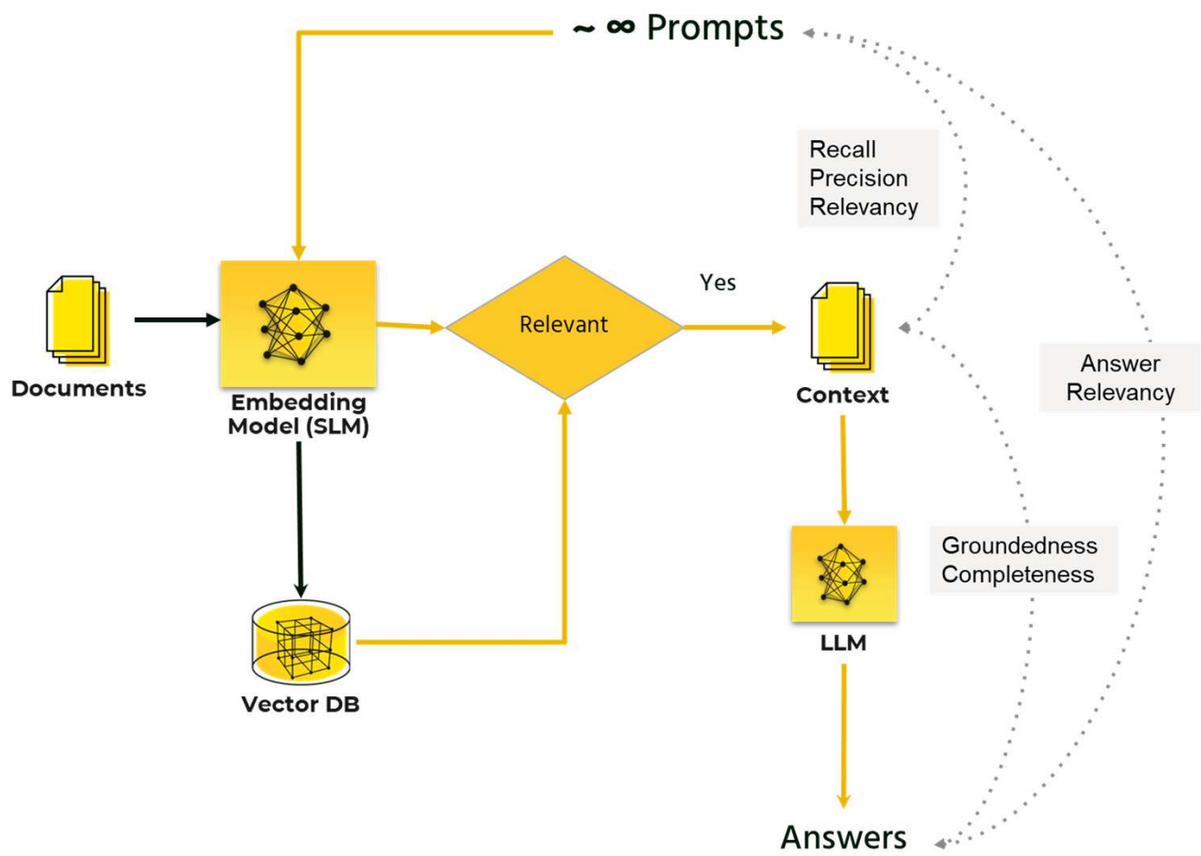
Finite collection of documents

Input Control: only allow prompts relevant to the knowledge base

Prompts should be bounded to those relevant to the knowledge-based from the documents in the vector database

Limited range of outputs

Controlling the set of documents and prompts results in limited set of outputs.



Usage Specific and Transparent Evaluation

Context-Specific Performance:

- Ensures the model is tested on queries relevant to the intended application.
- Avoids relying on generic benchmark that may not reflect real-world challenges.

Transparent and Explainable Metrics:

- Repeatable and reproducible evaluation
- Supports regulated industry by providing explainable, documented performance measures.
- Enables internal and external audits with traceable performance insights.
- Improves user confidence by making the decision-making process understandable.

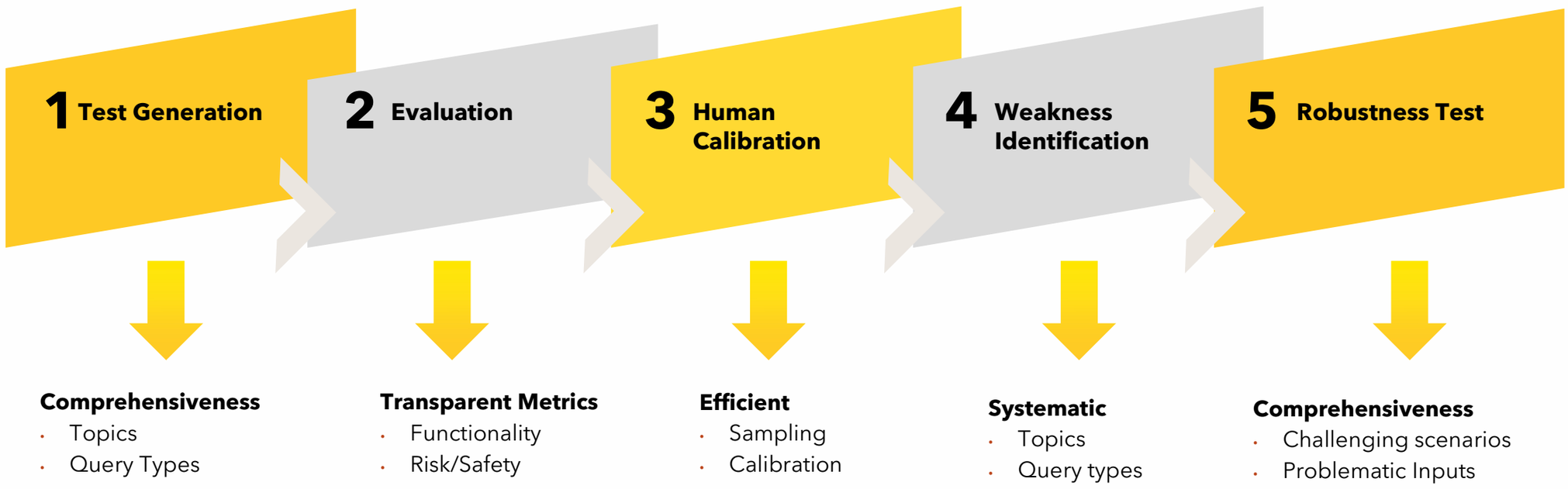
Continuous Improvement and Debugging:

- Simplifies the process of improving models by pinpointing weaknesses in model predictions.
- Makes model development more transparent, allowing for targeted updates and retraining efforts.

Model Validation is a Challenging Task

- **Unlike predictive AI, test cases have to be created**
- **How do we make sure test cases are comprehensive?**
  - How do we ensure comprehensive coverage?
  - How many and what kind of prompts?
- **How are we going to evaluate them?**
  - Multiple imperfect evaluation metrics.
- **Can we afford to create and perform comprehensive test?**
  - Can we automate and confidence with the outcome?

Model Validation Workflow and Tools



What Is Embedding?

An **embedding** is a representation of objects (such as number, words, sentences, or images) in a continuous vector space

- Embeddings translate complex, categorical inputs (like words) into numerical representations.
- These representations preserve relationships like similarity, context, and associations.
- The vectors typically have a fixed size (dimensionality), which is much smaller and more meaningful than the original input.

- Usage of Embedding in Model Validation
  - **Explainability**: Numerical vectors representation can be utilized for interpreting the model.
  - **Sampling**: Facilitate diverse and representative sampling by stratified sampling of embeddings
  - **Evaluation**: Useful to measure relevance/similarity measurements such as:
    - Whether the question is relevant to the retrieved chunks?
    - Whether the answer is relevant to the retrieved chunks?

Step 1: Test Generation

Query Generation for Testing

Comprehensive Queries

- Ensures comprehensive assessment of the RAG system's ability to retrieve relevant contexts and generate accurate answers for all topics.
- Enables targeted stress-testing and scenario-based testing of system capabilities such as reasoning, recall, and synthesis.

Diverse of Query Types

- Covers a broad range of system abilities, such as fact extraction, analytical reasoning, and handling ambiguity.
- Ensures robustness by testing under different linguistic formats (e.g., paraphrases, follow-up questions).
- Helps identify weaknesses in specific areas, such as handling noise, out-of-context queries, or numerical reasoning., ensuring the system similar yet differently phrased queries.
- Robustness Queries: Evaluates how well the system handles performs well under realistic and diverse user inputs.

Step 1: Test Generation

Automated Q&A Pair Generation

Document Chunks

Embeddings

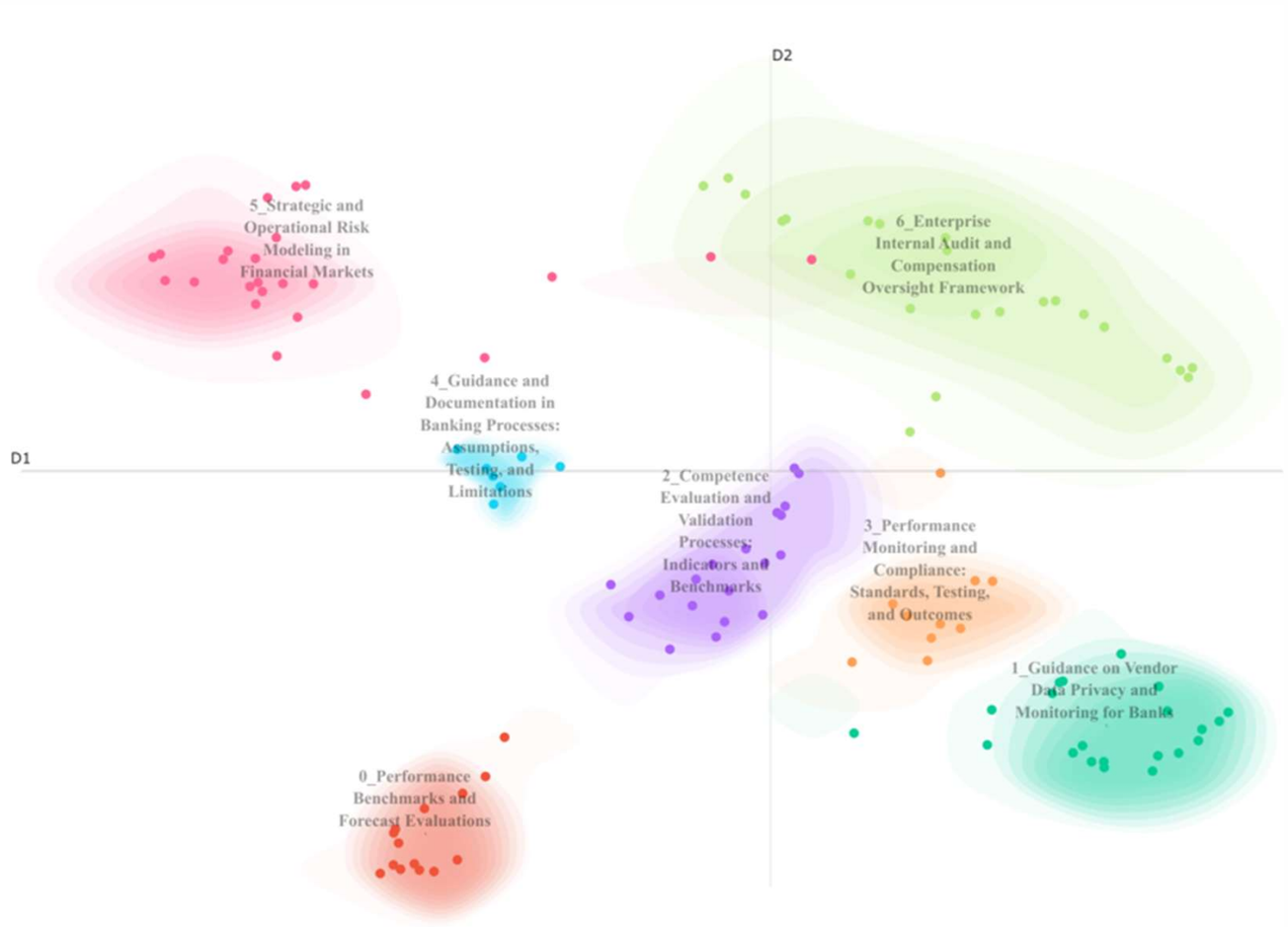
Dimensionality Reduction
Topic Clustering
Twinning Samples within Clusters

Stratified Sampling of Document Embeddings

Prompting

Automatic Q&A Pair Test Generation by LLM

Quality Control



Step 2: Evaluation

Evaluation Metrics

1. Functionality Evaluation:

Evaluate ability to retrieves, synthesizes, and generates information based on user queries

- **Retrieval** (Query → Retrieved Context): Evaluate ability to retrieve relevant instances
- **Generation**
  - (Context → Answer):
    - **Groundedness**: Evaluate ability to produce answers that is aligned with relevant context
    - **Completeness**: Evaluate ability to generate summary that captures all the critical information from context
  - (Query → Answer):
    - **Answer Relevancy**: Evaluate ability to provide answer that is aligned with user's query

2. Safety Evaluation

- **Toxicity**: Evaluate offensive, harmful, or inappropriate language.
- **Bias**: Detect and quantify demographic or sentiment bias.
- **Privacy**: Detect generation or leaking sensitive information.

Step 3: Human Calibration

Human Calibration via Conformal Prediction

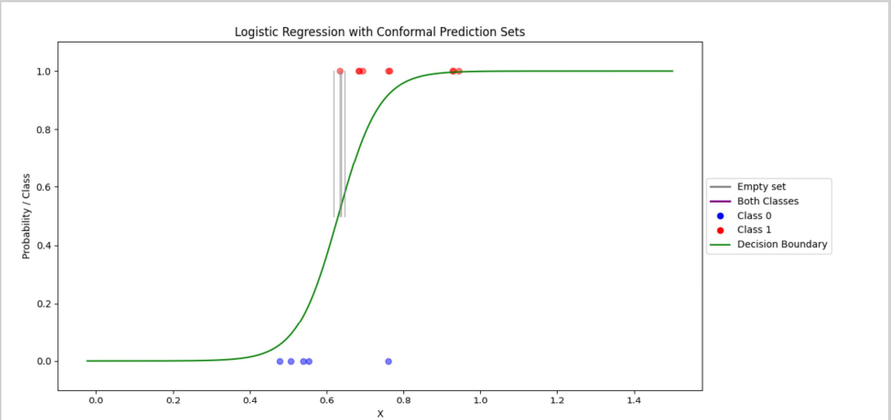
Bridging alignment between machine/algorithmic and human evaluation

- Efficient human labeling via sampling
- Statistical calibration
  - e.g., Logistic regression or isotonic regression
- Statistical confidence to make inference for unlabeled evaluation
  - Conformal Prediction

Calibrated measurement for deployment guardrails

Steps

- Samples of human labeled
- Calibrate machine evaluation vs. human labels
- Apply Conformal Prediction




```
Conformal Prediction Results:

Summary Table:
Prediction Type      Count      Percentage
-----------------------------------------
Empty sets           33         5.40%
Single class         578        94.60%
Both classes         0          0.00%
Score cutoff lower:  0.6109484114602697
Score cutoff upper:  0.6486293747057628
```

Step 4: Weakness Identification

Model Validation: Weakness Discovery

Pinpoint Areas of Low Performance:

- Identify topics, query types, or contexts where the model struggles (e.g., factual accuracy, reasoning tasks).

Analyze Interactions for Compound Weaknesses:

- Use bivariate analysis to detect combinations of factors (topics + query types) that lead to failures.

Cluster and Group Failures for Patterns:

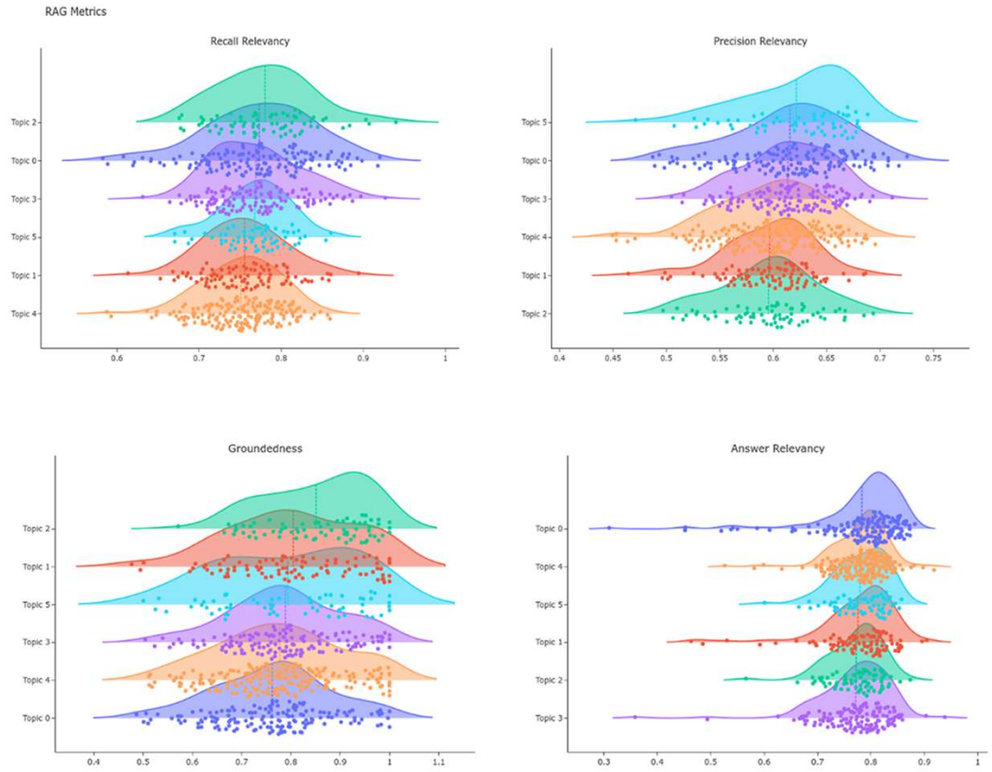
- Cluster failure cases (e.g., hallucinations, low precision) to detect recurring patterns and common failure modes.

Enable Targeted Improvements:

- Provide insights that guide targeted retraining, fine-tuning, or prompt design to strengthen weak areas.

Develop Risk Mitigation:

- Identify critical area for model monitoring and create guardrails for safe deployment.



Step 5: Robustness Test

Robustness in RAG System

Robustness in Retrieval-Augmented Generation (RAG) systems refers to the system's ability to consistently produce accurate, relevant, and coherent responses across a wide variety of input conditions, including adversarial, out-of-distribution, and noisy queries.

Dimensions of Robustness Testing

- **Adversarial Inputs**

  Test how well the RAG system handles deliberately misleading, ambiguous, or contradictory inputs.

- **Out-of-Distribution (OOD) Queries**:

  Test the system's performance on queries that are outside the domain of the training data or not well-covered by the document collection.

- **Input Variations (Perturbations and Noise)**: Evaluate the system's resilience to noisy, misspelled, or paraphrased queries.

Closing Remarks

Generative AI (GenAI) Capability and Challenge

- Excels in creative, open-ended scenarios to generate possible outcomes
- But, faces significant reliability challenges for high-risk applications

Rigorous Validation

- Comprehensive and Targeted Prompt Generation
- Contextual and Robustness Evaluations
- Transparent and Explainable Metrics
- Human Calibration via Statistical Methods
- Systematic Identification of Weaknesses

Essential Actions

- Adopt thorough validation frameworks to manage uncertainty effectively
- Implement robust guardrails for ethical and safe AI deployments
- Continuously refine and calibrate AI systems against real-world scenarios