Weakness and Robustness Evaluation

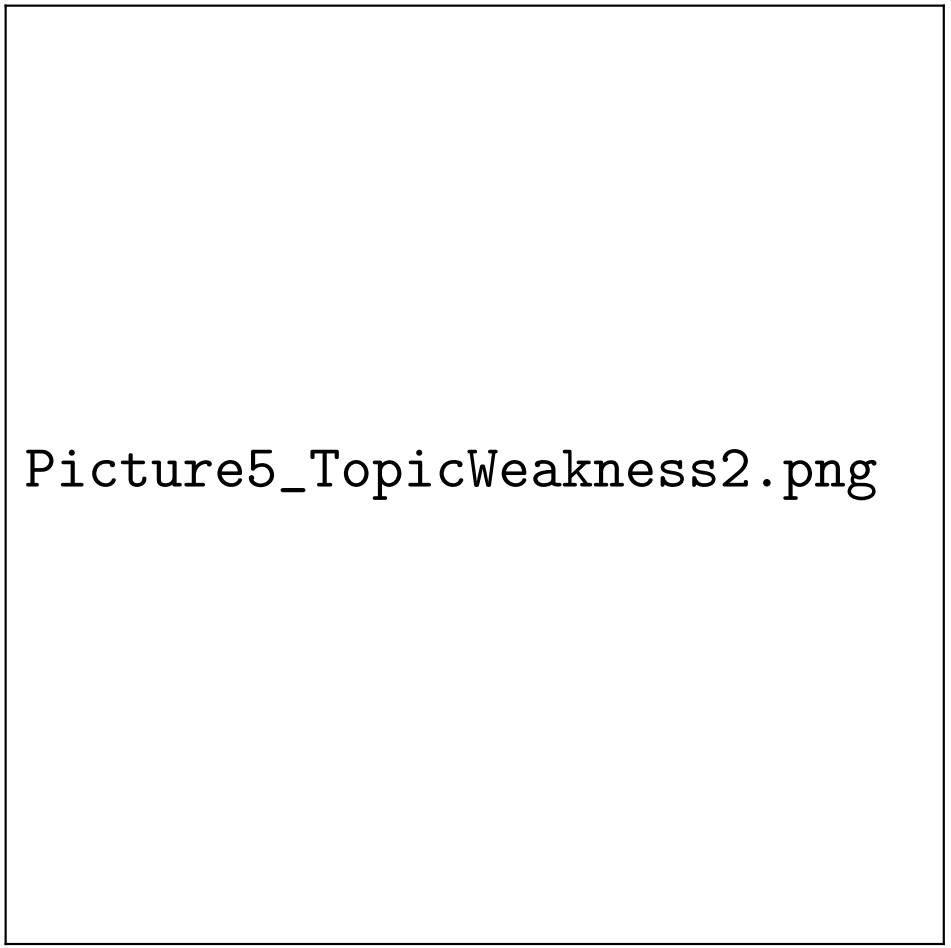
Objective

- Identify model weaknesses defined by the functionality and metrics
- Test robustness under challenging conditions.
- Guide targeted model improvements without degrading overall performance.
- Create actionable insights for retraining, prompt design, and guardrail for safe deployment.

Weakness Discovery

Methods:

- **Marginal Analysis**: Evaluate metrics per topic or query type.
  - Functionality metrics: Relevancy, Groundedness, Completeness, etc.
  - Query types
- **Bivariate Analysis**: Joint analysis of topic-query interactions.



Failure Mode Clustering

- Use embedding and distance matrix to group failures.
  - Queries
  - Context
  - Answers
- Use performance metrics as criteria.
- Use Random Forest for supervised clustering.
- Identify frequent failure patterns (e.g., groundedness, “why” questions).

Visualization Techniques

- **Marginal Analysis:** Distribution plots (e.g., groundedness, answer quality).
- **Bivariate Analysis:** Heatmaps across topic-query pairs.
- **Failure Clustering:** Visual summary of common modes.

Robustness in RAG Systems

Goal: Evaluate consistent performance under stress:

- **Adversarial Inputs**: misleading, ambiguous, or contradictory queries.
- **Out-of-Distribution Queries**: unfamiliar domain questions.
- **Noisy Inputs**: typos, grammar errors, paraphrases.

Query Types for Robustness Testing

- Ambiguous / Misleading / Vague
- Paraphrased / Contradictory / Edge Case
- Noisy / Out-of-Context / Implied Context
- Domain-Specific Jargon / Nonstandard Format / Multi-Intent

Sample System Prompts

- **Ambiguous:** Generate a question with multiple interpretations.
  - *System Prompt:* "From the given context, generate a query with ambiguous phrasing or multiple valid interpretations that could confuse the retrieval process."
- **Misleading:** Generate a question that contain false premises.
  - *System Prompt:* "Using the context, create a long and convoluted query that requires careful parsing to understand the question."
- **Vague or Underspecified:** Generate a question with with insufficient detail.
  - *System Prompt:* "Given the context, generate a vague or underspecified query that lacks critical details or specifics."

Sample System Prompts

- **Noisy:** Generate a question with typographical errors, incorrect grammar, or mixed languages.
  - *System Prompt:* "Given the context, create a query by introducing noise, such as typos, grammatical errors, or mixing another language into the question."
- **Out-of-Context:** Generate a question where information not present in the context.
  - *System Prompt:* "Based on the context, generate a query that deliberately asks about information not included in or related to the text."
- **Implied Context:** Generate a question to deduce implied or unstated context.
  - *System Prompt:* "From the given context, generate a query that relies on implied or unstated information to be understood."

Sample System Prompts

- **Paraphrased:** Generate a question phrased differently.
  - *System Prompt:* "Given the following context, generate multiple paraphrased versions of a query asking for the same information in different ways."
- **Contradictory:** Generate a question where information is contradictory.
  - *System Prompt:* "Using the context, create a query that contains contradictory statements, requiring the system to identify and address the conflict."
- **Edge Case:** Generate a question to to test rare or boundary conditions in the context.
  - *System Prompt:* "From the provided context, create a query about edge cases or scenarios that are unusual but theoretically possible."

Robustness Metrics: Adversarial Inputs

- **Error Rate on Adversarial Queries**: The percentage of incorrect responses to adversarial inputs.
- **Rejection Rate**: Percentage of cases where the system declines to provide an answer due to uncertainty.
- **Confidence Scores**: Measure how confident the system is when handling adversarial inputs.
- **Robustness Loss**: The increase in the system's error rate when adversarial inputs are introduced

Robustness Metrics: Out-of-Distribution (OOD)

- **Factuality Score**: Measure the factual accuracy of the system's response.
- **Relevance Score**: Determine whether the system retrieves the most relevant documents, even for unexpected queries.
- **Acknowledgment of Limitations**: Measure how often the system responds appropriately by stating that the requested information is unavailable.

Robustness Metrics: Input Variations (Perturbations and Noise)

- **Semantic Similarity**: Compare responses across different perturbations of the same query using cosine similarity of embeddings.
- **Response Stability Score**: Calculate the consistency in answers despite variations in query phrasing.

Key Takeaways

- Weakness detection is critical for trust and safety.
- Use compound analysis and clustering to find systemic issues.
- Evaluate robustness using diverse and challenging query types.
- Visualize results to guide corrective actions.