GroundGuard

ENSURING TRUSTWORTHY CHATBOT PERFORMANCE

Jenny Wei

Data Science
Senior Manager

weixue0932@gmail.com

01

Challenges and Design Approach

Why Groundedness Matters and How We Solve It

02

Solution Architecture

Building the GroundGuard System: Retrieval, Validation, Evaluation

03

Performance, Insights

How It Performs, What We Learned, and Future Enhancements

Challenges and Approach

Challanges

- Unexpected user query
- Unreliable Chatbot Responses
- Risk of Misinformation
- Need for Groundedness

Input Query Classification Using LLM or Custom Models

Classify user query based on the products, relevant topics, and edge cases.

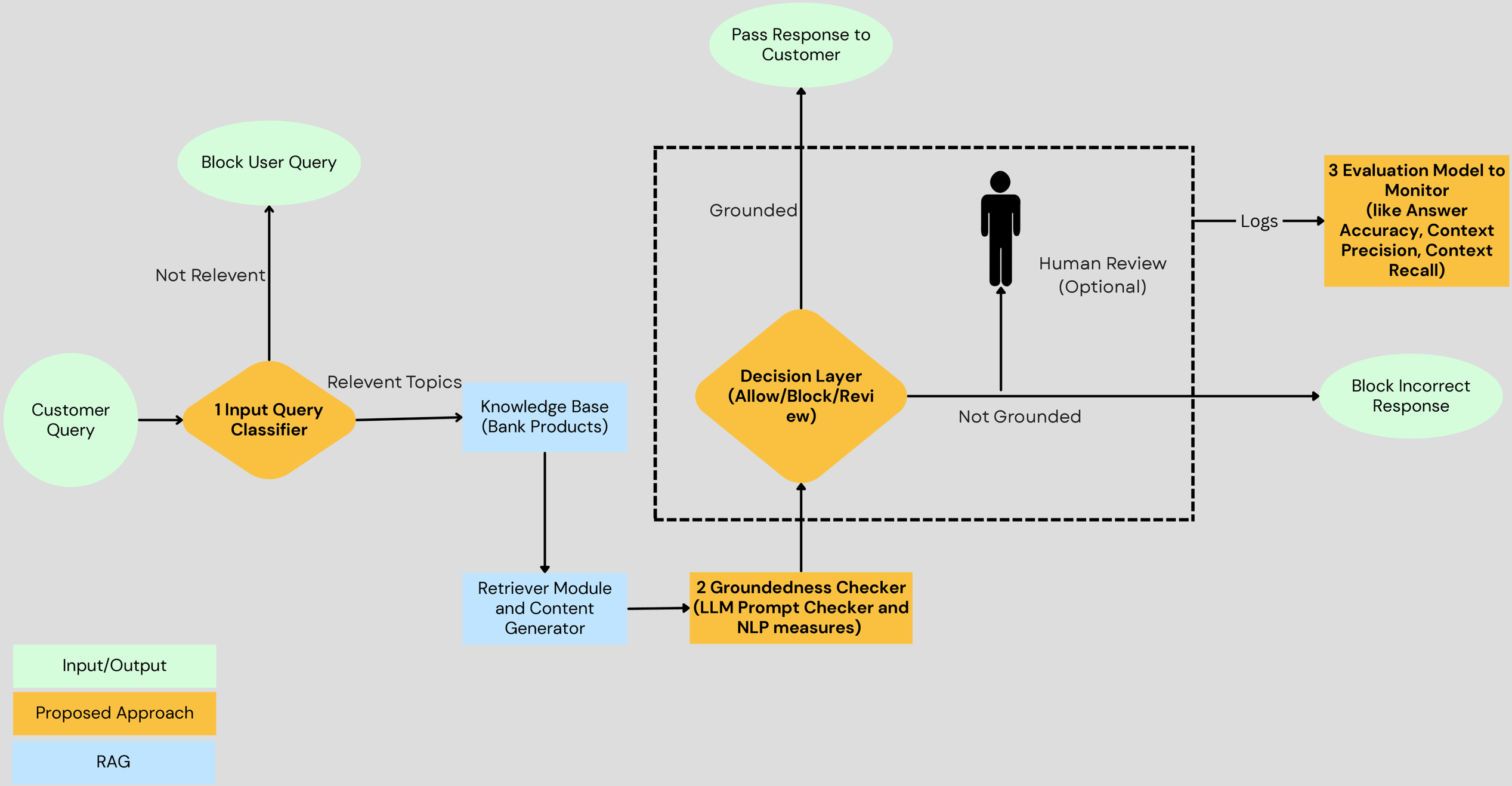
Response LLM-Driven Groundedness Check

Use a targeted LLM prompt and NLP measures to verify if the response matches retrieved knowledge.

Simple, Transparent, Reliable

Focused on enterprise-grade consistency, easy evaluation, and explainability

Let's visualize solution architecture



Input Query Classifier

- LLM Based Prompt to Classify User Query
- Bert-based Text Classifier

Prompt of Input Query Classifier using Tools

Classify the User query into one of the class list.
##Class list##: {class_type}
##Detail of the class list##: {class_detail}
##User query##: "{query}"
##Objective##: Classify the user query into one of the Class list.
##Detailed Instructions##:
1. Read the user query carefully.
2. Classify the user query into one of the class list.
3. If the user query does not match any of the class list, return "None".
##Response format##: {{"input_class": "class_name"}}

Fine-turned Model

- Based on pretrained open-source model distilbert/distilbert-base-uncased to classify user input query into classes {"product", "service", "harmful", "irrelevant", "generic", "sensitive", "ambiguous", "unanswerable"}
- Prepared human labelled sample data to fine-tune the model.
- Data size used for fine-tune model
  - Train (120 samples)
  - Validation (50 samples)
  - Test (56 samples)

Output Groundedness Checker

- LLM Based Prompt to Verify Answers Generated from the RAG solution
- Semantic Similarity Filtering

Prompt of Output Groundedness Checking using Tools

You are tasked with verifying whether an AI chatbot's response is grounded in the provided product knowledge base.

Definitions:

- "Grounded" means that factual claims, instructions, or key information in the response are supported by the provided context.
- "Ungrounded" means that the response introduces information that is missing, inconsistent, incorrect, or not explicitly supported by the context.

Please be extremely strict:

- If **important detail** is unsupported by the context, mark the response as "False".
- Minor paraphrasing that keeps the same meaning is acceptable.
- Speculation, assumptions, hallucination, or information beyond the context is not allowed.

Given:

- User query: {query}
- Product knowledge base context: {context_string}
- AI chatbot response: "{response}"

Task:

- Carefully read the chatbot response and the context.
- Compare the information closely.
- Answer "True" if **every part** of the response is grounded in the context; otherwise, answer "False".

Return only one word: **True** or **False**.

Semantic Similarity Filtering

- For each user query, calculate the semantic similarity score between the query and the chunks in knowledge base.
- If the max similarity score (cosine similarity) is less than 0.5, then the AI response is not grounded.

Decision Layer

Fully Automated
AI (LLM + simple rules)

For low/medium-risk interactions to maintain speed and scale

Human-in-the-Loop
AI flags + human reviewer

For critical topics (e.g., loans, compliance-sensitive replies) or initial rollout to build trust

Logging & Monitoring

All decisions (Pass/Block/Flag) are logged for audit, quality monitoring, and model retraining.

Evaluation Model to Monitor

Generation:

Faithfulness
Answer Relevancy
Answer Correctness
Faithfulness

Retrieval:

Context Precision
Context Recall

Traditional NLP Metrics:

BLUE
ROUGE
Semantic Similarity
Factual Correctness

Customer Metrics:

Rubics Based Scoring

Quantitatively and Qualitatively Evaluation

01

Evaluation results for input query classifier

LLM Prompt Model:
200+samples
Accuracy: 92%
Precision: 92%
Recall: 92%
F1-score: 91%

Fine-tuned Model:
Accuracy: 94.64%
Precision : 95.31%
Recall: 94.64%
F1:94.61%

- Human label ground truth data
- SME to review sample data and conduct analysis to include more edge cases
- Performance is only tested on sample data, the actual performance need to captured from the use of filters.

02

Evaluation results for output groundedness checker

LLM Prompt Model:
(19 samples)
Accuracy: 79%
Precision: 96%
Recall:79%
F1:85%

Semantic Similarity:
(19 samples)
Accuracy: 100%
Precision: 100%
Recall:100%
F1:100%

- Capture user's feedback to continue update the **prompt** of the Groundedness Checker
- SME to review the filtered AI response and improve the product and services documents

03

Monitor and Continuous improvements

- RAGAS Monitor Metrics are captured
- Combined with the input classifier
- and output filtering

- As a logging and monitor system, the input input query classification results, output groundedness checker, and RAGAS Metrics are captured to improve the models

01

Insight 1: Comprehensive evaluation

- For a robust enterprise-grade RAG system, a holistic evaluation combining faithfulness, relevance, precision, semantic similarity, and correctness is essential to reliably ensure grounded, trustworthy chatbot behavior.
- LLM-based classifier and groundedness checker are easy to build with high performance.
- Fine-tuned model needs more time for preparing data and hard to adapt for new data.

02

Insight 2: Selected Metrics

Metrics to build the GroundGuard

Input Query Class (LLM based)

Semantic Similarity

Groundedness Results (LLM based)

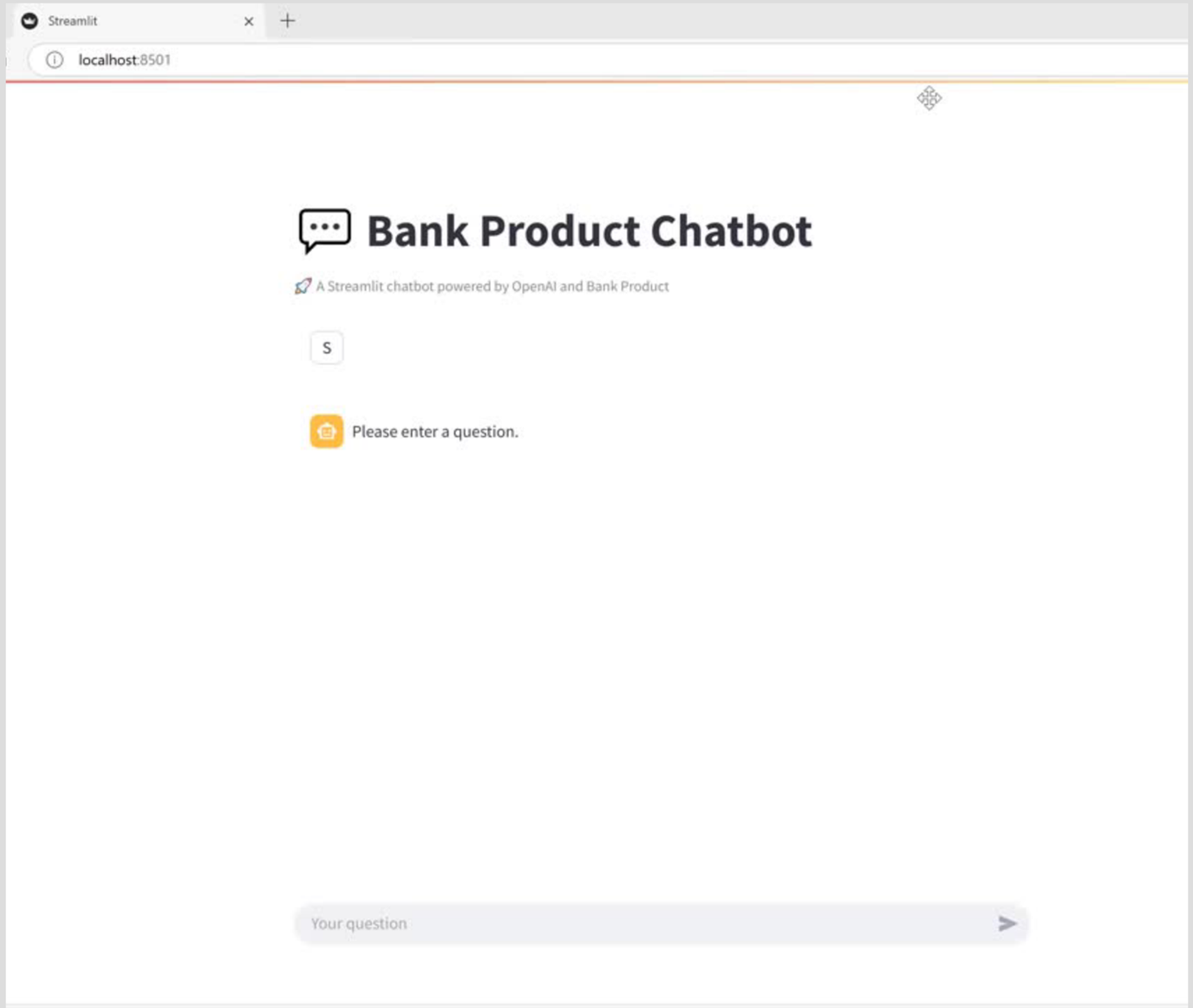
- Faithfulness (RAGAS)
- Answer Relevancy (RAGAS)
- Context Precision (RAGAS)
- Answer Correctness (RAG)

03

Insight 3: Tested with human feedback

- SME’s input and understanding of use cases are important. The definition of user input classifier and the output groundedness checking
- You can evaluate the model performance and continue to improve the results via monitor user’s feedback.

Demo



Next

01

More data

Need to test on real product and service data

02

Business cases

Need to work closely with SME to define the prompt and label ground truth data

03

Feedback loop

Need to build the feedback loop and monitor loop to record testing results from users.

Thank You

FOR YOUR TIME

Jenny Wei

Data Science
Senior Manager

weixue0932@gmail.com