Guardrails for LLM Chatbot Applications – Ensuring Output Groundedness

Enhancing Factual Accuracy & Safety in Conversational AI

Dr Sruthy Skaria
0478581406
sruthyscaria@gmail.com

Agenda

- Introduction to Guardrails in LLM Applications
- Why Output Groundedness Matters ?
- Key Components of Guardrails
- Implementation Strategies & Examples
- Evaluation Metrics & Testing
- Case Studies & Best Practices
- Next Steps
- References & Q&A

Introduction to Guardrails

- **Definition:** Guardrails are layers of checks and controls that ensure LLM outputs are factually accurate, safe, and compliant.
- **Importance:**
  - Reduces misinformation
  - Increases trust and reliability
  - Meets regulatory and ethical standards
- **Use Cases:** Banking, healthcare, legal, customer support

Why Output Groundedness Matters

- **Groundedness:** The degree to which an answer is directly supported by an approved knowledge base or reference data.
- **Key Points:**
  - Prevents hallucinations (i.e., invented or unsanctioned details)
  - Ensures factual accuracy by strictly using provided data
  - Critical in high-stakes domains (e.g., banking)
- **Example:**
  - Reference Data: "Missing mortgage payments can result in late fees and foreclosure."
  - Incorrect Answer: "Missing mortgage payments may lead to legal actions." (Not grounded)

Components of Guardrails

- **Input Filtering:**
    - Sanitizes user input to prevent injection attacks and harmful content.
- **Output Verification:**
    - **Groundedness Check:** Compares generated response with reference data.
    - **Safety Filter:** Checks for toxic language, profanity, bias, etc.
- **Custom Actions:**
    - Define domain-specific policies (e.g., no financial advice)
- **Monitoring & Logging:**
    - Real-time tracking of outputs for continuous improvement

Implementation Strategies

LLM-Based Verification:

- Use a combined prompt to extract a groundedness score and safety scores

Deep Evaluation Guardrails:

- Multi-layered evaluation (automated tests, human-in-the-loop, continuous monitoring).
- Adaptive thresholds & detailed reporting (e.g., toxicity, bias, defamation).

Nemo:

- An advanced AI framework for building conversational agents.
- Utilizes deep evaluation techniques for robust safety and quality.
- Supports real-time monitoring and continuous improvement in AI outputs.

Implementation Strategies cont.

Guardrail AI:

- An AI system designed to enforce safety, neutrality, and professionalism in responses
- Implements strict guardrails for toxicity, profanity, sensitive topics, and bias
- Uses deep evaluation methods to ensure ethical and accurate chatbot behavior.

Example Flow:

1. User query received by the chatbot
2. Input guardrail
3. Chatbot generates an answer
4. Guardrail system checks: - Scope of this task
   - **Groundedness:** Does the answer derive strictly from approved reference data?
   - **Safety:** Is the language safe, neutral, and compliant?
5. Final output is determined based on both checks

Groundedness Evaluation Methodology

- **Definition:** An answer is 100% grounded if it is entirely derived from the reference data using the exact modal language
- **Strict Requirement:** Even if generally correct, any deviation or extra details marks it as not fully grounded.
- **Prompt Example:**
  - "Evaluate if the answer strictly adheres to the reference data without introducing any new details."
- **Decision Logic:**
  - If groundedness score ≥ 0.7 and safety thresholds met → Valid output.
  - Else, provide appropriate fallback messages.

Evaluation Metrics & Testing

Metrics

- **Accuracy:** Percentage of correct responses.
- **False Positives/Negatives:** Incorrectly accepting or rejecting responses.
- **Score Distribution:** Groundedness and safety scores.
- **Testing Process:**
    - Generate test cases using defined scenarios (Valid, Safety Fail, Moderation Fail, Both Fail).
    - Compare final outputs to expected results.
    - Use dashboards and visualizations (e.g., histograms, bar charts) to analyze performance.

Usecase – Banking Chatbot

- **Scenario:** A customer asks, “What happens if I miss a mortgage payment?”

- **Reference Data:** “Missing mortgage payments can lead to late fees and foreclosure.”

- **Guardrail Outcome Examples:**

    - **Valid:** "Missing mortgage payments can lead to late fees and foreclosure."
    - **Moderation Fail:** "Missing payments may result in legal action." (Extra unsanctioned info)
    - **Safety Fail:** "Missing payments can lead to late fees, but the bank might sue you." (Unsafe advisory language)
    - **Both Fail:** "Missing payments can result in lawsuits and severe penalties." (Both unsanctioned info and unsafe)

- **Learning:** Strict adherence to reference data is critical.

Custom LLM-Based Guardrails – Approach & Rationale

- **Overview of Our Approach:**
  - We built a custom LLM-based guardrail system rather than using NVIDIA NeMo Guardrails.
  - This approach allows seamless integration with any chatbot, regardless of the development team's framework.
  - It gives us full control over prompt design and decision logic.
- **Key Principles:**
  - **Strict Groundedness:**
    - Answers must be entirely derived from approved reference data.
    - Answers must use the exact modal language from the reference data (e.g., if "may" is used, not "could" or "can") without adding any new details.
  - **Safety Enforcement:**
    - No advisory language, guesses, or opinions are permitted.
    - The system checks for toxicity, profanity, bias, defamation, and sensitive topics.

Custom LLM-Based Guardrails – Approach & Rationale cont.

- **Why This Matters in Banking:**
  - Banking communications must be factually accurate and comply with regulatory standards.
  - Strict guardrails help avoid misleading or unsafe responses.
  - Custom guardrails allow us to tailor thresholds specifically for the sensitive nature of financial advice.
- **Our Custom vs. NeMo Guardrails:**
  - **NeMo Guardrails** is a robust, DSL-based system—but requires a commitment to the NeMo ecosystem.
  - **Our Custom Solution** provides flexibility and integration independence for any chatbot framework.

Suggested Thresholds, Conditions

Groundedness:

- **Threshold:** ≥ 0.7 & **Rationale:** Ensures that answers are well supported by reference data; any extra, unverified detail results in rejection

Toxicity:

- **Threshold:** < 0.3 & **Rationale:** Banking messages must be free of offensive, hateful, or discriminatory language.

Profanity:

- **Threshold:** 0 & **Rationale:** Any presence of swear words or vulgar language triggers rejection.

Bias:

- **Threshold:** < 0.3 & **Rationale:** Responses must remain neutral and objective; even slight bias is unacceptable.

Defamation:

- **Threshold:** < 0.3 & **Rationale:** Prevents language that could defame or disparage others, ensuring a fair and professional tone.

Sensitive Topics:

- **Threshold:** < 0.3 & **Rationale:** Ensures that responses do not inadvertently reference sensitive subjects such as violence, self-harm, or explicit content.

Neutral_and_Balanced_Tone:

- **Threshold:** ≥ 0.8 & **Rationale:** Ensures that responses are delivered in a balanced and objective manner, avoiding overgeneralizations or speculative language.

Professional_Language:

- **Threshold:** ≥ 0.8 & **Rationale:** Guarantees that the language used is formal, respectful, and adheres to industry standards, thus avoiding misinformation or unprofessional tones.

Implementation Details

- **Implementation Notes:**
  - Our guardrail system leverages LLM calls to evaluate both groundedness and safety.
  - The combined prompt returns a JSON with all the scores.
  - Decision logic uses these scores to determine if an answer is valid or should be rejected.
- **References & Industry Practices:**
  - **Google Perspective API:**
    - Default toxicity thresholds ~0.7, but banking requires stricter (~0.3).
  - **OpenAI Content Guidelines:**
    - Emphasize tailored moderation in regulated industries.
  - **Financial Regulatory Standards:**
    - Impose strict requirements on accuracy and neutrality in communications.

Final Note

- These thresholds are a starting point.
- Pilot testing, user feedback, and compliance reviews will further refine these values to ensure regulatory compliance and optimal customer experience.

Best Practices

- **Continuous Monitoring:**
  - Regularly update your knowledge base.
  - Monitor and refine guardrail thresholds.
- **Human-in-the-loop:**
  - Use human review for ambiguous cases.
- **Iterative Improvement:**
  - Use feedback from deployment to enhance guardrail performance.
- **Compliance:**
  - Ensure alignment with legal and regulatory requirements.

Next Steps

- Multiple Prompts
- Storing things in config
- Automation of testing
- Building chatbot
- Moving chatbot to configs so that we can change llm easily
- Optimising for time and cost -business evaluation
- A-B Testing with users
- Testing with other guardrail solutions - NeMo guardrails, Guardrails - AI, and DeepEval guardrails
- Manual review and testing

References

- LLM Guardrails: Your Guide to Building Safe AI Applications
- NeMo Guardrails Keep AI Chatbots on Track | NVIDIA Blogs
- Top 20 LLM Guardrails With Examples | DataCamp
- How to implement LLM guardrails | OpenAI Cookbook
- LLM Guardrails for Data Leakage, Prompt Injection, and More - Confident AI
- LLM Guardrails: A Comprehensive Guide to Securing AI Applications | by awesomesaras | Medium
- NVIDIA NeMo Guardrails GitHub: https://github.com/NVIDIA/NeMo-Guardrails
- OpenAI API Documentation: https://platform.openai.com/docs/api-reference/introduction
- Related Research on LLM Safety and Groundedness:
    - Bender, E. M., & Koller, A. (2020). "Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data."
    - Marcus, G., & Davis, E. (2020). "GPT-3, Bloviator: OpenAI’s language generator has no idea what it’s talking about.“