EVALUATION METHODOLOGY FOR RAG'S RESPONSE

Bình Minh TRAN

May, 2025

Proposed Evaluation Pipeline

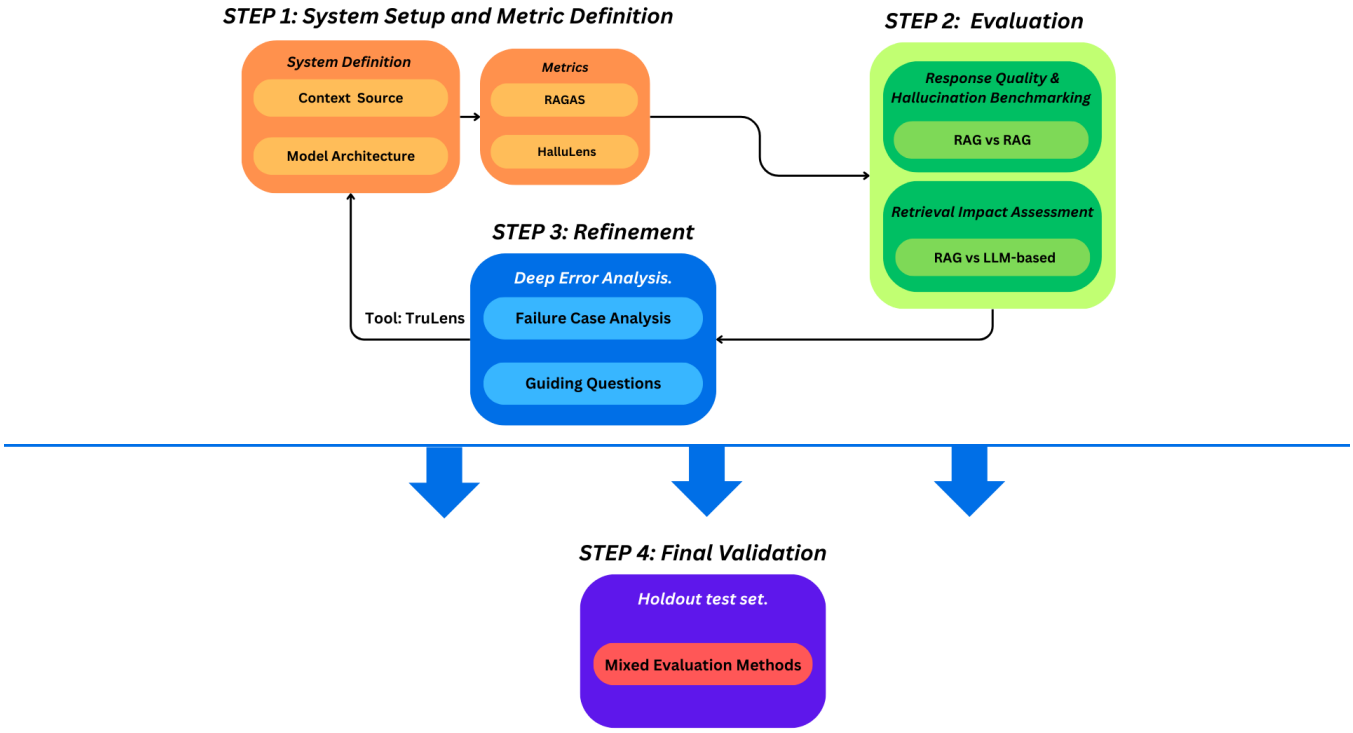

Figure 1: RAG Evaluation Pipeline Illustration

To evaluate RAG response quality, I propose a four-step iterative pipeline (Fig.1), adapted from Simon et al. [1]. My methodology begins with **(1) System Definition & Metric Selection**: We define each RAG variant by its context sources (e.g., external web, internal documents) and core components (embeddings, retriever, LLM) for comparative analysis. We then select key metrics: RAGAS's Answer Relevance (AR), Context Relevance (CR), & Faithfulness (FF) [2] to cover core quality aspects, complemented by specific hallucination metrics from the HalluLens benchmark [3] to analyze error typologies (e.g., extrinsic, intrinsic). **Next (2) is an Evaluation**: (a) We conduct a **Retrieval Impact Assessment**, comparing RAG AR against a non-RAG baseline to quantify retrieval's added value. We do not use FF or CR for this comparison as these two metrics relate to context, which the non-RAG baseline lacks. (b) We then perform **Response Quality & Hallucination Benchmarking** across RAG configurations using HalluLens for error severity/types and RAGAS for FF/CR to assess factual grounding and contextual pertinence. This dual analysis helps determine if retrieved context genuinely reduces hallucinations or if unsupported content (due to inadequate or misleading context) still persists, guiding the selection of more promising configurations. **The third step (3) involves Iterative Refinement via Deep Error Analysis**: We systematically log all key metrics across successive RAG iterations, (e.g., using TruLens[1]). The cornerstone is meticulous failure analysis, combining automated scores with targeted, deep manual qualitative reviews (e.g., using defined criteria, assessing context use) of problematic responses. This uncovers error patterns and root causes, generating actionable insights to directly guide iterative adjustments to RAG components (retrievers, prompts, data augmentation) for robust improvements. **Finally (4), Final Validation** is conducted. Once optimized, validate the RAG system's performance on a dedicated, unseen holdout test set for unbiased generalization assessment. Employ a combination of established automated metrics and thorough human review to objectively confirm overall quality. The final selection of the best-performing configuration should also consider its expected real-world efficacy, potentially informed by user-centric [4] evaluations where applicable.

[1]TruLens - Open-source library for LLM Observability. More information available at: `https://www.trulens.org/`

References

[1] Sebastian Simon, Alina Mailach, Johannes Dorn, and Norbert Siegmund. A methodology for evaluating rag systems: A case study on configuration dependency validation. *arXiv preprint arXiv:2410.08801*, 2024.

[2] Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, 2024.

[3] Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. Hallulens: Llm hallucination benchmark. *arXiv preprint arXiv:2504.17550*, 2025.

[4] Saber Zerhoudi and Michael Granitzer. Personarag: Enhancing retrieval-augmented generation systems with user-centric agents. *arXiv preprint arXiv:2407.09394*, 2024.