<u>Test case: 1</u>

Question: “What’s your plan to win a Quidditch match?”

Character: Draco Malfoy

1. LLM (Language Model)

RAG Output:

- The prompt and model combination produce a concise, in-character answer.
- Draco’s tone is smug, references Slytherin, a fancy broom, and boasts about beating Gryffindor.

Agentic RAG Output:

- LLM output is more generic, focusing on "skillful play, teamwork, strategy."
- Lacks Draco’s trademark arrogance and book-specific rivalry, despite traits provided.

Improvement Suggestions:

- Use stronger prompt engineering for agentic LLM calls, requiring explicit reference to Draco’s competitive and condescending style.
- Consider instruction-tuning or few-shot in-character examples.
- Use larger or newer models if possible for more nuanced voice.

2. Agent Orchestration (Tool Use)

RAG Output:

- The pipeline clearly retrieves context and traits, then feeds them into the LLM.
- Output is clearly grounded and evaluated.

Agentic RAG Output:

- Agent calls `GetCharacterTraits`, but response does not leverage traits fully.

- Sometimes skips `SearchContext` or does not combine both tools before responding.

- LLM prompt includes traits but still produces generic output.

Improvement Suggestions:

- Force agent to always retrieve both traits and context, then synthesize both into the prompt.

- Clean up and de-duplicate trait information before prompt injection.

- Use memory or a planning step to ensure multi-tool reasoning.

3. Prompt Engineering

RAG Output:

- Prompt is explicit: "NEVER break character", "smug, competitive, arrogant", "concise", "a hint of arrogance".

- Context passages are included, and instructions request Draco’s voice.

Agentic RAG Output:

- Prompt is less directive; allows for generic responses.

- Lacks dynamic emphasis on rivalry, Draco’s book-specific taunts, or his signature phrases.

Improvement Suggestions:

- Add more in-character sample outputs to the prompt.

- Require the output to reference rivalry, Slytherin, or Draco’s family.

- Summarize or highlight the most relevant context sentences for the prompt.

4. Retrieval

Both:

- FAISS and SBERT embedding retriever are used.
- RAG output seems to surface more relevant, Draco-specific Quidditch content.
- Agentic output sometimes uses context that is tangential, not tightly focused on Draco's Quidditch ambitions.

Improvement Suggestions:

- Enhance retrieval by combining dense and keyword (BM25) search.
- Tune chunk size and overlap for better alignment with character scenes.

5. Evaluation

Both:

- Automated scoring for relevance, authenticity, and context.
- RAG output gets full marks; Agentic RAG gets marked down for character authenticity.

Improvement Suggestions:

- Use the LLM for self-evaluation: "Does this sound like Draco? Is it grounded in context?"
- Allow for paraphrased context matches, not just verbatim.

<u>Test Case: 2</u>

Question: "What would you say to someone bullying a friend?"

Character: Harry Potter

. LLM (Language Model)

- Current: You are using an open-source LLM (e.g., Llama-3 or similar) via Hugging Face Transformers.
- **Observation:**
  - The model generates relevant and in-character answers when given strong prompts and context.
  - However, sometimes the output falls back to generic patterns or fails to tightly ground answers in the retrieved passages.
- **Improvement:**
  - Model Fine-Tuning: Fine-tune the LLM on Harry Potter dialogues and situations to make responses more authentic and context-sensitive.
  - Try Newer Models: If possible, experiment with larger or more recent models (e.g., Llama-3-8B, Mixtral, OpenHermes, Mistral) for richer, more nuanced generation.

---

2. Agent/Tool Orchestration

- **Current:**
  - The agent uses a tool-based framework (likely LangChain Agents) for stepwise reasoning (GetCharacterTraits, SearchContext).
  - Actions and observations are explicitly logged.
- **Observation:**

    - The agent sometimes stops after retrieving traits, not always chaining both trait and context retrieval before generating a final answer.
    - Tool outputs (especially character traits) are verbose and sometimes repetitive.
- **Improvement:**
    - Tool Chaining Logic: Enforce that both GetCharacterTraits and SearchContext must always be called before answer generation.
    - Agent Memory: Use conversation memory to allow for more multi-turn, context-aware dialogue.
    - Tool Output Formatting: Clean up outputs from tools for easier downstream prompt integration (avoid repetition in traits, use concise keys).

---

3. Prompt Engineering

- **Current:**
    - The prompt is explicit about role-playing, character traits, and use of book context.
    - Instructions are clear about using first-person pronouns, book themes, and avoiding out-of-world language.
- **Observation:**
    - The prompt is effective but could be more dynamic in inserting example scenarios and retrieved passages.
    - Sometimes, the LLM is not sufficiently grounded, especially if the context isn't relevant or is too long.
- **Improvement:**

- Dynamic Prompting: Use templates that insert the most relevant context snippets and traits, and summarize long context to the most pertinent sentences.
- Few-Shot Examples: Add a few in-character sample Q&A pairs to the prompt to guide the LLM's style and voice.
- Explicit Guardrails: Add instructions that the model must reference or paraphrase the retrieved context in the response.

---

4. Retrieval Stack

- **Current:**
  - FAISS + SBERT for dense passage retrieval.
- **Observation:**
  - Retrieval works, but sometimes the context is too broad or not directly about bullying or Harry's reaction to bullies.
- **Improvement:**
  - Hybrid Retrieval: Combine dense (vector) and sparse (keyword/BM25) retrieval for more targeted context.
  - Chunking Strategy: Experiment with chunk sizes and overlap to ensure responses are always tightly grounded in relevant passages

---

5. Evaluation/Guardrails

- **Current:**
  - Automatic evaluation of relevance, authenticity, and context accuracy.
- **Observation:**
  - Sometimes context accuracy fails because the response is not strictly found in context, even if it's in-character and accurate.

- **Improvement:**
  - Contextual Validation: Use more flexible validation (allow paraphra sing, not just exact matches).
  - Self-Evaluation: Prompt the LLM to self-check its answer for grounding in the provided context.

<u>Test Case:3</u>

Question: How would you react to seeing a ghost for the first time?

Character: Ron Weasley

This output has relevance and strong character authenticity like nervous, humorous and has low contextual accuracy but not fully supported by retrieved passage.

Rag model produces a detailed in-character answer and showing ron's nervous humor. But Agentic RAG output is generic and not in Ron's unique voice. To improve this with strong prompt engineering LLM calls.

To improve the retrieval part with hybrid search (dense + BM25) and better chunking.