AI Dungeon Master

An Adaptive Storytelling Agent with Long-Term Memory

Team Name: NotDecided

Team Members: Aryan Chakravorty
Naveen

Abstract

This report presents the system architecture and technical design of the *AI Dungeon Master* — an adaptive storytelling agent designed to emulate the creativity and continuity of a human Dungeon Master in role-playing games. The system combines a stateful graph architecture using **LangGraph**, long-term vector memory through **ChromaDB**, and **an adaptive Retrieval-Augmented Generation (RAG)** pipeline powered by **Google's Gemini** model.

Traditional tabletop storytelling depends on a human Dungeon Master to weave dynamic narratives, manage player actions, and maintain world coherence. Large Language Models (LLMs), while capable of vivid text generation, often fail to preserve long-term context and consistency. To address this, our project introduces an AI-driven framework that maintains narrative memory, tracks player choices, and delivers persistent, evolving storylines — ensuring a coherent and immersive role-playing experience.

1 System Architecture

The AI Dungeon Master is built on a modular architecture orchestrating several key technologies. **LangGraph** serves as the central state machine, managing the flow of data between the user, the generative model, and the memory stores. **ChromaDB** provides persistent long-term memory, while the **Gemini Pro** model handles all core language tasks, including narration, summarization, and routing.

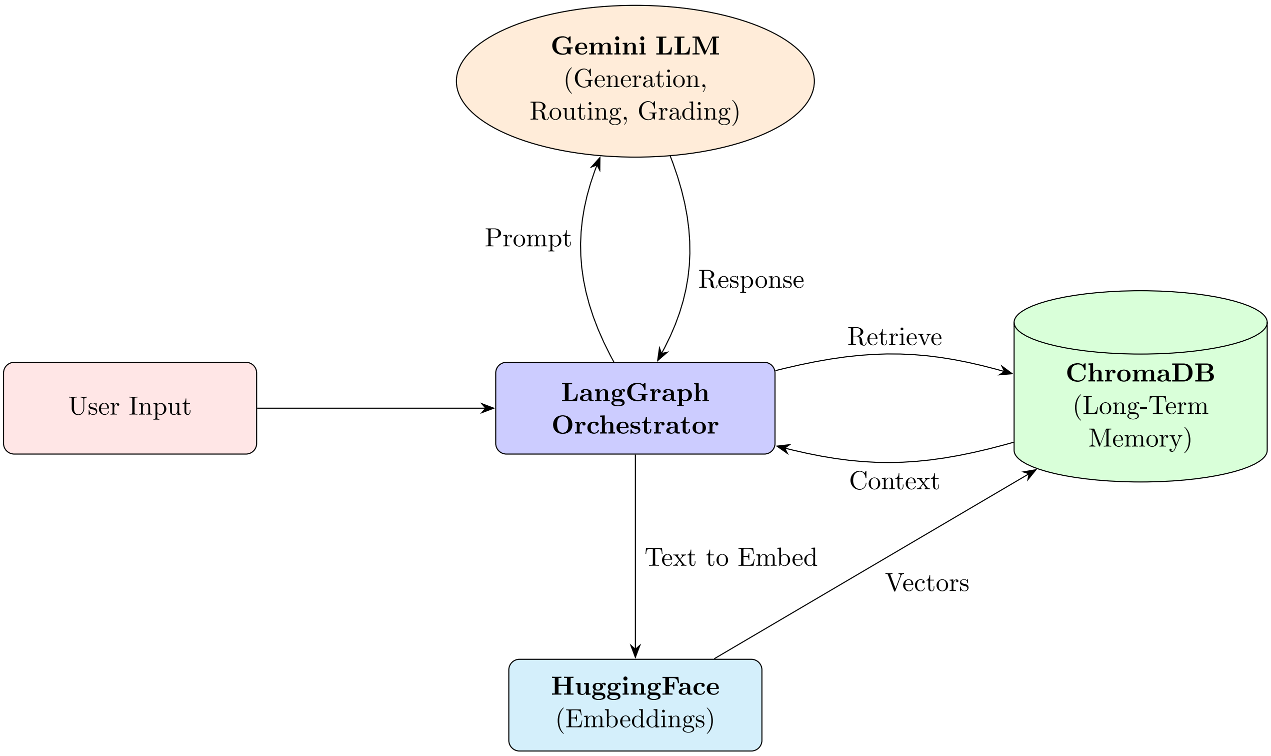

Figure 1: High-level system architecture showing the interaction between core components. LangGraph orchestrates all logic.

2 Memory and Data Flow

The system employs a dual-layer memory approach to ensure both short-term conversational context and long-term narrative persistence. The entire process is governed by an Adaptive RAG pipeline that intelligently decides when and how to use retrieved memory.

2.1 Adaptive RAG Memory Workflow

User inputs are first processed for memory storage. The `store_facts_to_chroma` function summarizes the user's action into a concise fact, which is then embedded and stored in **ChromaDB**.

When generating a response, the `chat_node` rewrites the user's query for optimal retrieval, fetches relevant documents, and passes them through a series of LLM-based grading steps to ensure relevance and prevent hallucination, before generating the final narrative output.

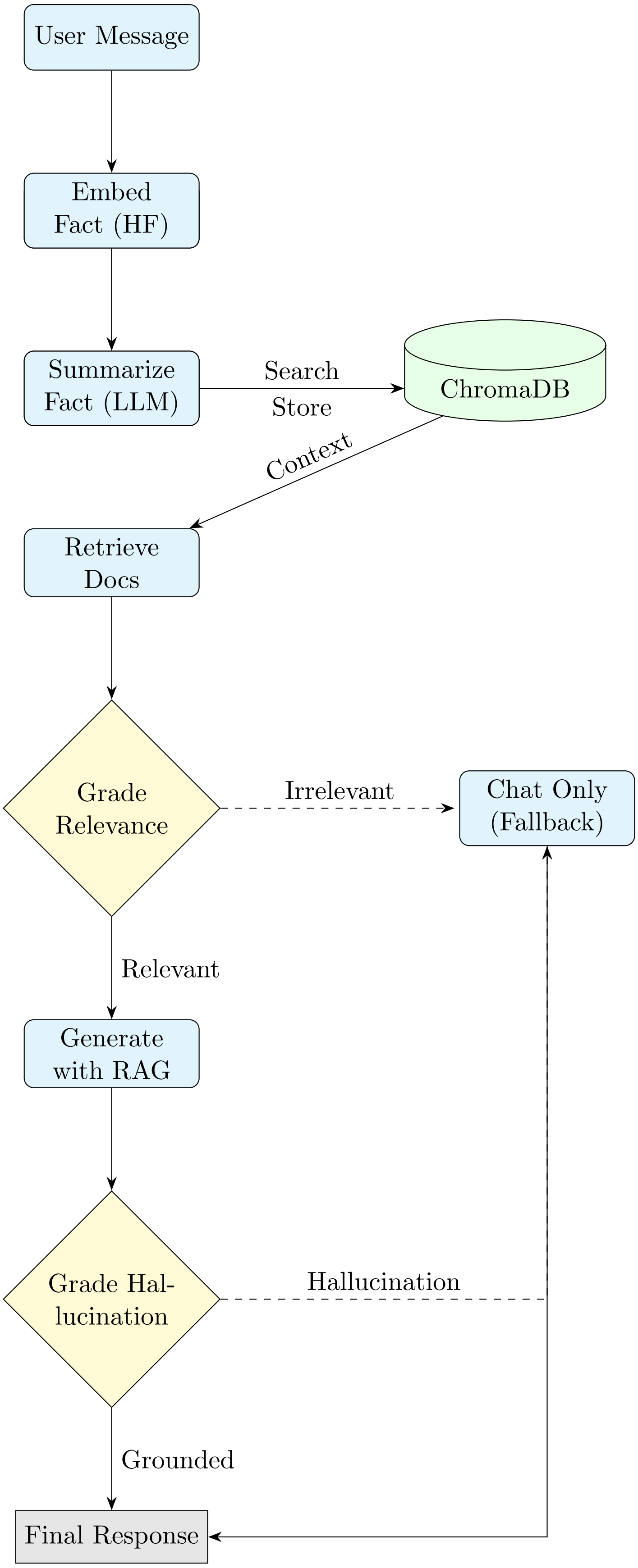

Figure 2: The Adaptive RAG memory workflow, from storage to graded retrieval and generation.

3 LangGraph Workflow

The agent's logic is modeled as a state graph, which ensures a predictable and robust execution flow. The graph begins by storing facts from the user's latest input and then proceeds to the main chat logic. The chat node can conditionally call external tools (e.g., Tavily Search) if required by the LLM, before looping back to generate a final response.

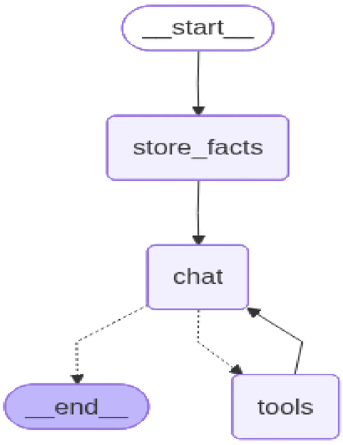

Figure 3: The state machine defined in LangGraph. The flow proceeds from storing facts to the core chat logic, with a conditional branch for tool usage.

4 Evaluation Metrics (Placeholder)

The system's performance is evaluated against key storytelling and technical criteria. The Adaptive RAG pipeline is specifically designed to improve memory relevance and reduce hallucinations compared to a standard RAG or chat-only approach.

Resources

- https://supermemory.ai/blog/3-ways-to-build-llms-with-long-term-memory/
- https://langchain-ai.github.io/langgraph/tutorials/rag/langgraph_adaptive_rag/

Table 1: System Performance on a Long-Form Narrative Test.

Metric / Category	Score (%)	Key Evidence / Description
Long-Term Event Memory	94%	30+ turn recall consistently
Major NPC Memory	98%	Shepherd: 3 encounters, perfect continuity
Multi-Encounter Tracking	97%	Cross-referenced all meetings
Specific Dialogue/Quotes	93%	Recalled specific phrases like ”Setback,” insults, and prophecies
Emotional Arcs	96%	Tracked character trajectory: Rage → doubt → anguish
Item Tracking	96%	All items consistent (journal, whistle, amulet, locket)
Location Consistency	95%	All locations remained stable and coherent throughout the narrative
Lore Integration	97%	Connected texts → prophecy → character motivation
Story Payoffs	98%	Successfully executed long-term payoffs (Locket mystery, redemption arc)
Quest Continuity	92%	Amulet quest line was maintained correctly across 37 turns
Minor NPC Memory	85%	Minor NPCs (Raven, Elsworth) were recalled when prompted
Player Recap Handling	68%	Failed to generate a complete summary under a heavy context load at Turn 31
Interrogative Retrieval	94%	Direct questions from the player triggered excellent and relevant recall
Proactive Synthesis	90%	Showed unprompted integration of past events in Turns 34 and 37