Gloombot

Gloomhaven: Jaws of the Lion chatbot

Marcus Eklund

Supervisor: Antonio Prgomet

2025-08-12

This paper presents a chatbot assistant for Gloomhaven: Jaws of the Lion leveraging a Retrieval-Augmented Generation (RAG) framework that integrates a large language model with a domain-specific vector database constructed from the game's glossary. Various chunking methods and similarity search metrics—Squared L2, Inner Product, and Cosine Similarity—were evaluated to optimize the retrieval of relevant context for generation. Results demonstrate that paragraph-level chunking aligns well with the glossary's structure, and Cosine Similarity yields higher-quality responses despite increased computational cost. The system was evaluated using the RAG Triad framework, confirming strong context relevance, groundedness, and answer relevance, thereby minimizing hallucination risk. Integration with Discord enabled multi-platform accessibility during live playtesting, where the chatbot effectively facilitated rule clarifications in real time. These findings illustrate the efficacy of RAG architectures for rule-based assistant applications in complex board games.

Table of contents

1 Introduction

Modern board games frequently incorporate complex rule systems, evolving scenarios, and extensive reference materials that can challenge even experienced players [Machuqueiro and Piedade, 2023]. Gloomhaven: Jaws of the Lion is a prominent example of such a game, combining cooperative campaign play with detailed rule interactions and scenario-specific conditions [Fandom, 2022]. While the game's learn-as-you-play design aims to make the experience more accessible, players still often need to consult glossaries and reference guides to clarify specific mechanics during gameplay. This process can interrupt the game flow and reduce player engagement [Zagal, 2011].

Recent advances in natural language processing have enabled the development of intelligent assistants capable of retrieving and generating contextually relevant information in response to user queries [Jurafsky and Martin, 2025]. Retrieval-Augmented Generation (RAG) combines the strengths of semantic search and generative language models to provide grounded, accurate responses sourced from structured knowledge bases [Lewis et al., 2020]. When integrated into conversational interfaces, such systems can offer immediate, reliable assistance to players seeking clarification on complex rule interactions.

This paper presents the design, development, and evaluation of a chatbot assistant for Gloomhaven: Jaws of the Lion. The system integrates a vector-based retrieval pipeline with generative models to answer rule-related questions and is deployed via Discord to support seamless interaction across devices. To ensure reliability and minimize hallucinations, the assistant's outputs were evaluated using the RAG Triad framework, which assesses context relevance, groundedness, and answer relevance [TruEra, 2024].

The following sections describe the theoretical background of the game and retrieval-augmented generation, the methodological approach used to build and test the system, and the results of playtesting and evaluation. This work demonstrates the potential of retrieval-augmented chatbots to enhance the tabletop gaming experience by reducing rule lookup time and improving player confidence in rule interpretations.

2 Theory

This section presents the theoretical foundations relevant to the design and development of the chatbot assistant.

2.1 Gloomhaven: Jaws of the Lion

Gloomhaven: Jaws of the Lion is a standalone cooperative campaign-based board game based on the rules and setting of Gloomhaven published by Cephalofair Games. It features a prequel campaign set before the events of its predecessor in the city of Gloomhaven with 25 scenarios for 1-4 players to play through as four unique classes designed to be compatible with other games in the Gloomhaven series. It is intended to provide an easier introduction to the series and includes a simplified learn-to-play-guide instead of a rule book that walks players through its first five scenarios, gradually introducing new rules and game concepts in a 'learn-as-you-play' manner. Consequently, it features some rules omissions compared to other titles in the series. [Fandom, 2022]

The game's layered rule system introduces new mechanics gradually, but over time, players must consult multiple sources (scenario guides, character ability cards, and reference glossaries) to clarify interactions and special conditions. This can create cognitive overhead and frequent interruptions to gameplay, especially for new players. As a result, tools that can help quickly retrieve and explain rule details are valuable for maintaining game flow and reducing ambiguity.

2.2 Chatbot

A chatbot is a software application designed to simulate human conversation. In recent years, advances in natural language processing (NLP) have greatly improved chatbot capabilities, enabling them to interpret user queries and generate coherent, contextually appropriate responses [Jurafsky and Martin, 2025]. Chatbots are often deployed to support users in retrieving information, completing tasks, or learning complex systems such as tabletop games.

The development of an effective rules-assistant chatbot requires both accurate retrieval of relevant information and the capacity to express that information in a clear, conversational

form. To this end, retrieval-augmented generation techniques were employed [Lewis et al., 2020].

2.2.1 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a framework that combines information retrieval with natural language generation to produce accurate and contextually grounded responses. In this approach, user queries are first processed to retrieve relevant content from a reference corpus, and the retrieved content is then passed to a generative language model that formulates the final answer.

Retrieval in RAG systems commonly relies on vector similarity search, in which text passages and queries are embedded into dense vector representations in a high-dimensional space. Embeddings are numerical vectors that capture semantic meaning by positioning similar texts near each other in this space. The similarity between query and document vectors can be measured using various metrics, such as:

- **Squared L2:** Measuring the absolute geometric distance between vectors.
- **Inner Product:** Focusing on vector alignment and magnitude.
- **Cosine Similarity:** Measuring only the angle between vectors, ignoring magnitude.

These methods enable efficient retrieval of semantically relevant information even when the query phrasing differs from the source material [Karpukhin et al., 2020, Reimers and Gurevych, 2019, ChromaDB, 2024].

State-of-the-art generative language models, such as large transformer architectures pretrained on diverse text corpora, are commonly used to synthesize responses conditioned on the retrieved evidence. This retrieval-augmented approach helps mitigate the problem of hallucination by grounding model outputs in authoritative reference texts.

2.2.2 Evaluation

Evaluation of chatbots involves assessing multiple dimensions of performance to ensure the system is reliable, accurate, and helpful [Deriu et al., 2020]. Common evaluation criteria include:

- **Accuracy:** Whether responses are factually correct and consistent with authoritative sources.
- **Relevance:** Whether the response directly addresses the user's query and intent.
- **Clarity and Fluency:** Whether the answer is expressed in clear, natural language that is easy to understand.
- **Response Time:** The speed with which the system retrieves and generates an answer.

- **User Satisfaction:** The degree to which users find the interaction helpful and pleasant [Li et al., 2016].

Evaluation can be performed through a combination of automated metrics and human assessments. Automated approaches often measure retrieval quality, correctness, and text fluency, while manual evaluation relies on expert review or user studies to gather qualitative feedback.

2.2.2.1 RAG-Triad

For retrieval-augmented generation systems, a more specialized evaluation framework known as the RAG Triad has been proposed to systematically assess performance and reduce the risk of hallucination. The RAG Triad comprises three complementary evaluation dimensions:

- **Context Relevance:** This dimension verifies whether the retrieved documents or context chunks are relevant to the user's query. Since the retrieval step determines what information the language model can use, irrelevant context increases the likelihood that unrelated or incorrect details are incorporated into the response. Evaluating context relevance ensures the system is grounded in appropriate evidence.
- **Groundedness:** Even with relevant context, language models can produce responses that go beyond the retrieved facts or embellish them in ways that sound plausible but are unsupported. Groundedness evaluation involves identifying individual claims within the generated response and checking whether each claim can be traced back to the retrieved materials. This step helps ensure factual alignment between the answer and its supporting evidence.
- **Answer Relevance:** Finally, the response itself must still be helpful and directly responsive to the original question. Answer relevance assesses whether the generated text actually addresses the user's intent and provides useful information, rather than simply rephrasing context or including extraneous details.

By achieving satisfactory evaluations across all three dimensions of the RAG Triad, developers can make a well-supported claim that the system is hallucination-resistant—up to the limits of the knowledge base it retrieves from. In other words, if the underlying corpus is accurate, a RAG system that passes these evaluations is also likely to be accurate [TruEra, 2024].

This structured approach allows developers and evaluators to more precisely identify weaknesses in retrieval, grounding, or relevance and to systematically improve the reliability of retrieval-augmented generation applications.

3 Method

This section describes the development process, system architecture, and evaluation procedures used to create the chatbot assistant for Gloomhaven: Jaws of the Lion.

3.1 Data Collection and Preprocessing

The primary source of reference material for the chatbot was the official Gloomhaven glossary. This glossary was collected and prepared for retrieval by dividing the text into smaller units, or "chunks," suitable for embedding and similarity search. Different chunking strategies were explored to balance retrieval granularity and contextual completeness. Variations in chunk size were also tested to determine the optimal configuration for accurate query resolution.

3.2 Embedding and Vector Store

After preprocessing, each text chunk was transformed into a dense vector representation using embedding techniques. These embeddings encode semantic information to enable effective similarity comparison. The resulting vectors were stored in a ChromaDB vector database, which supports efficient retrieval of the most relevant passages given a user query.

During development, multiple similarity search methods were evaluated to identify the approach that provided the highest retrieval precision and best supported answer generation.

3.3 Answer Generation

Once relevant context chunks were retrieved from the vector store, they were combined with the user query to form a prompt for answer generation. This step was performed using OpenAI's language models, which synthesized responses conditioned on the retrieved content. This retrieval-augmented generation (RAG) approach was chosen to help ground outputs in authoritative rule text and reduce the risk of hallucination.

3.4 Discord Integration

To facilitate easy access across devices and enable real-time interaction during gameplay, the chatbot was integrated with Discord. This integration allowed users to issue queries and receive responses through a familiar chat interface. The Discord bot was configured to handle incoming messages, retrieve context from ChromaDB, generate answers with the language model, and post responses back to the appropriate Discord channel.

3.5 Evaluation Procedure

The system was evaluated using both automated and manual assessments. In particular, the RAG Triad framework was applied to measure:

- **Context Relevance:** Whether retrieved chunks were pertinent to the query.
- **Groundedness:** Whether generated responses were faithful to the retrieved context.
- **Answer Relevance:** Whether responses adequately addressed the original question.

These evaluations helped identify and address areas where retrieval or generation performance could be improved.

3.6 Playtesting

Finally, the chatbot was used during an actual Gloomhaven: Jaws of the Lion play session to observe its performance in a real-world scenario. User interactions during gameplay provided practical insights into response quality, retrieval effectiveness, and overall utility in supporting game flow.

4 Results

This section presents the results from the development and evaluation of the chatbot assistant. Key areas of focus include the effectiveness of different chunking strategies, the performance of various similarity search functions, and the quality of generated responses as assessed using the RAG Triad framework. In addition, observations from a live gameplay session are reported to illustrate the assistant's performance in a real-world setting.

4.1 Chunking Strategies

Three chunking strategies were tested for preparing the Gloomhaven: Jaws of the Lion glossary:

- **Fixed-size chunking (1000 characters):** Produced chunks that were too broad, often including irrelevant or disjointed information.
- **Semantic chunking:** Generated small, focused chunks, but many lacked the necessary context to support comprehensive answers.
- **Paragraph-based chunking:** Resulted in contextually cohesive and complete chunks, with minimal irrelevant content.

Among the three, paragraph-based chunking provided the most complete and useful context for retrieval, based on manual inspection of retrieval results and generated answers.

4.2 Similarity Search Metrics

Three distance functions were evaluated for retrieving the most relevant chunks from the vector database. The evaluation was conducted by prompting the chatbot with a controlled query and measuring:

- Response time (latency)
- Completeness of the answer

In Table 4.1 we can see that Squared L2 is the fastest and Cosine Similarity the slowest however in Table 4.3 we can see that Cosine Similarity gives the most complete answer.

4.3 RAG Triad Evaluation

The chatbot's outputs were evaluated using the RAG Triad framework, which scores answers based on:

- **Context Relevance:** Whether the retrieved chunks were relevant to the query.
- **Groundedness:** Whether the generated answer was supported by the retrieved context.
- **Answer Relevance:** Whether the response was helpful and addressed the user's intent.

Manual scoring of a test set of user queries yielded the following average results in Table 4.5. These results indicate that the system generally retrieves and generates responses that are accurate and contextually grounded, with minimal hallucination.

4.4 Play Session Observations

The chatbot was tested in a live gameplay session of Gloomhaven: Jaws of the Lion. During the session:

- The assistant was accessed via Discord for ease of use across multiple devices.
- We (the players) used the bot to clarify rule ambiguities in real time.
- Queries were answered within a few seconds.
- We (the players) were able to play a full session without having to open the glossary once.
- A few queries failed due to overly vague phrasing or missing glossary coverage but with slight tweaking to the queries the bot was able to give helpful answers.

The playtest demonstrated the practical usefulness of the chatbot and confirmed its ability to support smooth gameplay with minimal disruption.

Table 4.1: Time per distance calculation (Query: What is the turn order?)

Distance function	time(ms)
Cosine similarity	1691
Squared L2	1343
Inner Product	1554

Table 4.3: Aswers per distance calculation (Query: What is the turn order?)

Distance function	Answer
Cosine similarity	The turn order in Gloomhaven is determined by the initiative value of each character and monster set. At the start of every round, two cards are selected by each character, with one chosen as the initiative card. The initiative card is then revealed, and the initiative order is arranged from low to high initiatives. If there's a tie between a character and a monster, the character goes first. If there's a tie between two characters, the tie breaker is the initiative value of the second card. If two or more characters are long resting, they can choose the order of the long rests. When it's a character's turn, based on the initiative value of their initiative card, both cards are played, one after the other in either order, with one being used for its top action and the other being used for its bottom action.
Squared L2	The turn order in Gloomhaven is determined by the initiative value of each character and monster set. At the start of every round, two cards are selected by each character, with one chosen as the initiative card. The initiative card is then revealed, and the initiative order is arranged from low to high initiatives. If there's a tie between a character and a monster, the character goes first. If there's a tie between two characters, the tie breaker is the initiative value of the second card. If two or more characters are long resting, they can choose the order of the long rests.
Inner Product	The turn order in Gloomhaven is determined by the initiative value of each character and monster set. At the start of every round, two cards are selected by each character, with one chosen to be the initiative card. The initiative card is then revealed and the initiative order is arranged from low to high initiatives. If there is a tie between a character and a monster, the character goes first. If there is a tie between two characters, the tie breaker is the initiative value of the second card. If a character declares a long rest instead of selecting cards, their initiative value is 99.

Table 4.5: Average RAG Triad scores

RAG Triad Dimension	Score (out of 5)
Context Relevance	4.6
Groundedness	4.3
Answer Relevance	4.7

5 Conclusions and Discussion

This section summarizes the key findings from the project, discusses their implications, and outlines potential improvements for future work. The discussion is organized into four themes: chunking strategies, similarity search performance, RAG Triad evaluation, and practical usage during gameplay.

5.1 Effectiveness of Chunking Methods

Paragraph-level chunking emerged as the most effective approach due to the glossary's inherent structure. This method maintained the semantic integrity of individual entries without introducing irrelevant information, unlike fixed-size chunking which either fragmented concepts or included excess content. Semantic chunking, while precise, produced chunks too small to capture necessary context. These findings highlight the value of tailoring chunking strategies to the natural divisions in the source material to optimize retrieval and response quality.

5.2 Similarity Search Performance

Testing different distance functions showed a trade-off between speed and completeness. Squared L2 distance was the fastest to compute, but Cosine Similarity consistently retrieved the most relevant chunks, resulting in more accurate and complete answers.

Because the glossary entries were generally short and self-contained, small differences in vector similarity could have a greater impact on retrieval quality. Cosine Similarity's ability to measure the direction of vectors rather than just their magnitude proved particularly beneficial for matching the semantic meaning of concise glossary terms, even when exact wording differed from the query.

Given that correctness and clarity were prioritized over minimal latency in this use case, Cosine Similarity emerged as the preferred method despite its slower computation time.

5.3 RAG Triad Evaluation

Using the RAG Triad framework to assess context relevance, groundedness, and answer relevance provided a comprehensive view of the chatbot's reliability. Positive evaluation across all three metrics indicates that grounding the language model's responses in retrieved content effectively minimizes hallucinations, enhancing trustworthiness. This structured evaluation approach proved valuable for validating the chatbot's performance in a domain requiring precise and accurate information.

5.4 Live Gameplay Session

Feedback from a live play session further validated the system's practical utility. Players found the chatbot helpful in resolving rule uncertainties, and its availability via Discord made it easy to use across different devices during gameplay. The bot was able to reduce interruptions, allowing players to focus more on the game itself. However, some limitations were also observed, including failures on vague queries and situations where there was a scenario-specific rule that was not included in the glossary.

5.5 Future Work

The chatbot's knowledge was limited to a single glossary source, potentially restricting its comprehensiveness and ability to handle all rule nuances. Evaluation was also constrained by a limited number of play sessions. Future work could incorporate additional official rule materials, support multimodal inputs such as images, and fine-tune models on domain-specific text to improve accuracy and versatility. Expanding evaluation to include user feedback and error analysis would further inform development and refinement.

5.6 Summary

This project demonstrates the viability of using retrieval-augmented generation frameworks to build effective, context-aware chatbots for complex tabletop games. By carefully aligning knowledge chunking, similarity search methods, and thorough evaluation, it is possible to deliver accurate and timely assistance that enhances the player experience. While challenges remain, especially in expanding knowledge coverage and optimizing performance, this work lays a strong foundation for future development of intelligent game assistants and other domain-specific language model applications.

Appendix

Here are some of the queries posed to Gloombot in discord. Figure 5.1 shows how it can't answer a scenario-based question since the glossary does not contain the scenarios but tells you us where we can find the information. In Figure 5.2 and Figure 5.3 we can see how Gloombot responds to a question about something that is not a part of Gloomhaven: Jaws of the Lion or completely outside of the game. In Figure 5.4 and Figure 5.5 we see how Gloombot gives accurate and complete information to questions about aspects contained in Gloomhaven: Jaws of the Lion.

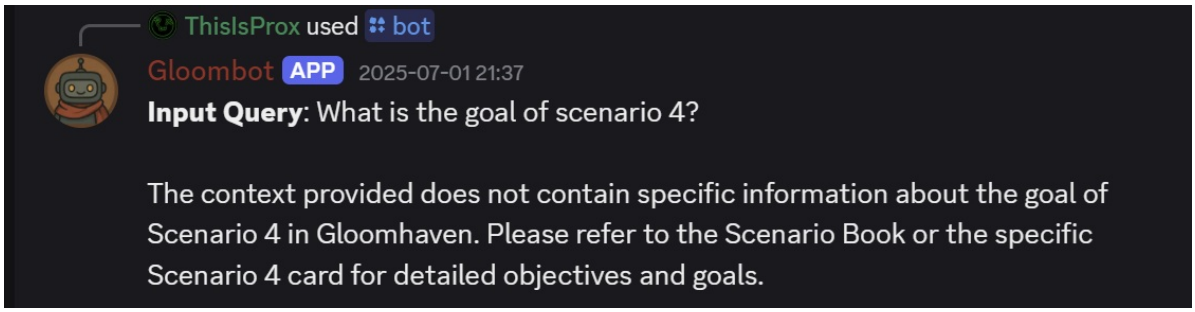

Figure 5.1: Scenario-based question for gloombot on discord

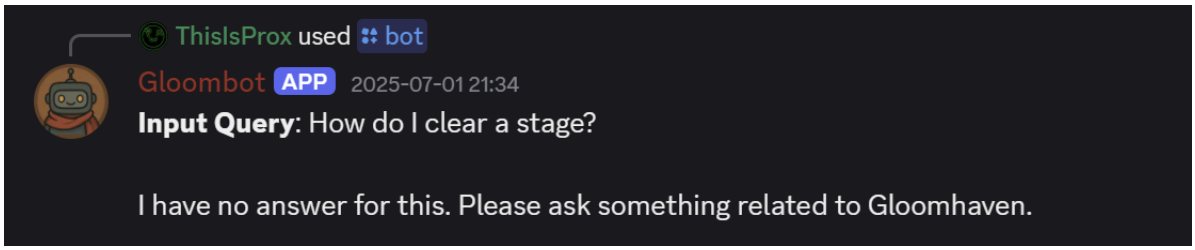

Figure 5.2: Question about clearing a stage which the game does not contain

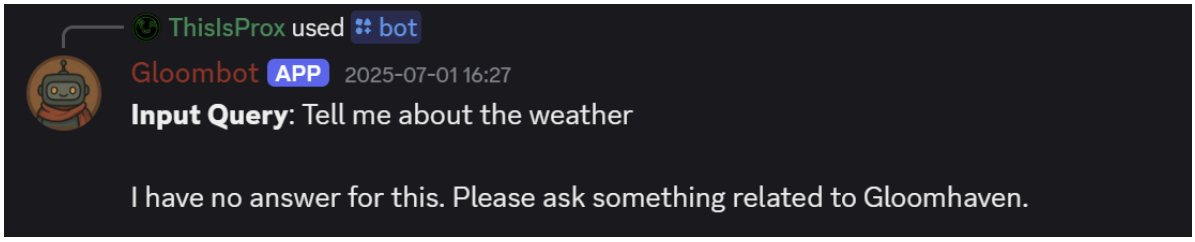

Figure 5.3: Question about the weather

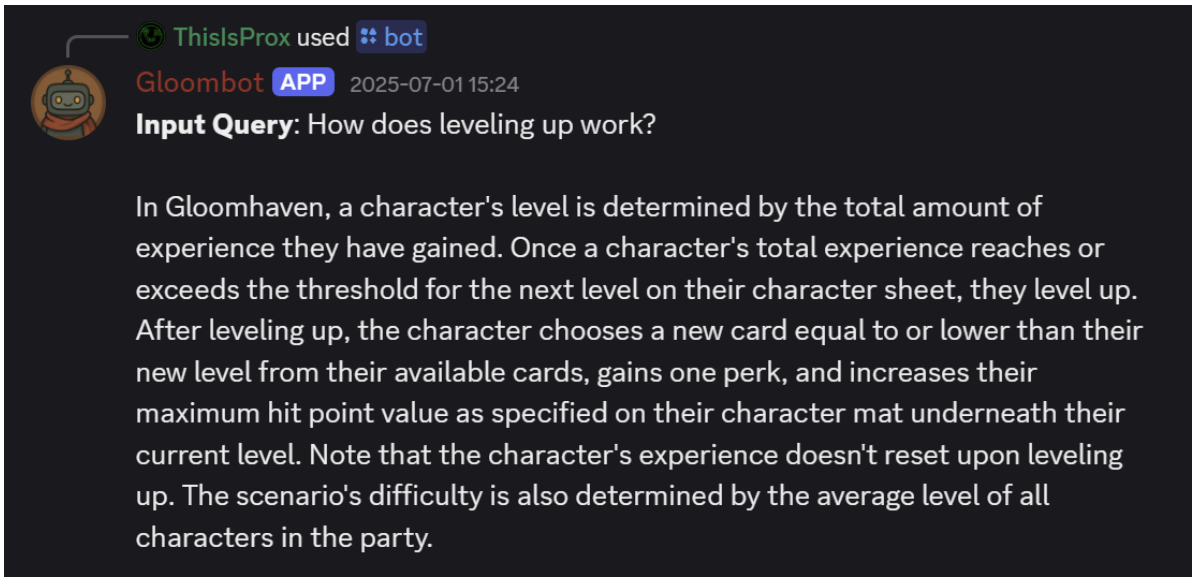

Figure 5.4: Question about how to level up

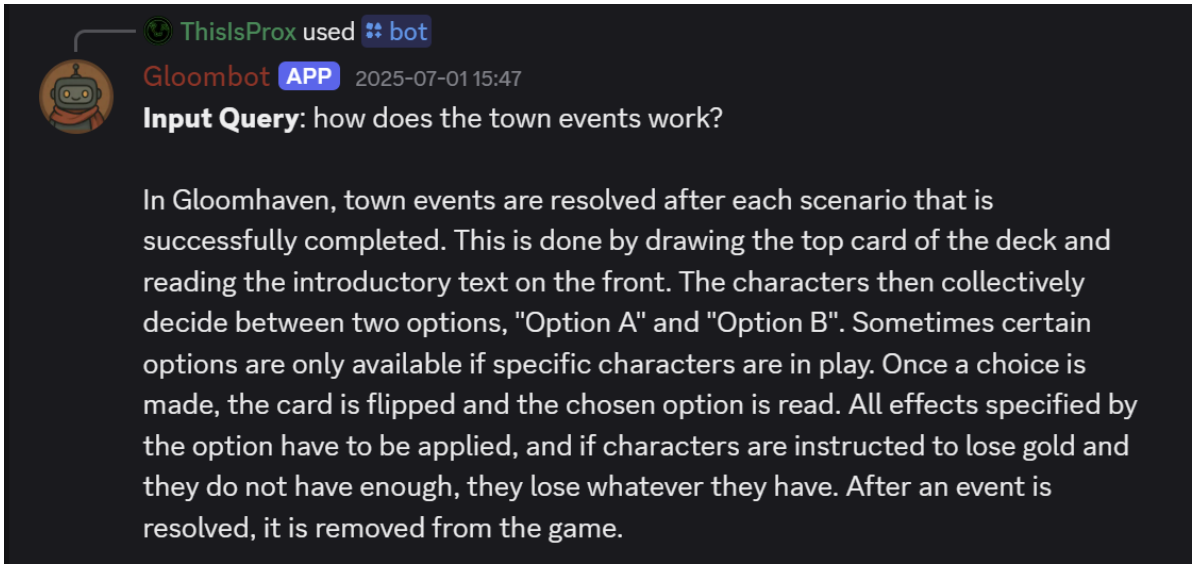

Figure 5.5: Question about town events

References

ChromaDB. HNSW index configuration, 2024. URL https://docs.trychroma.com/docs/collections/configure#hnsw-index-configuration.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, pages 755–810, 2020. URL https://arxiv.org/pdf/1905.04071.

Fandom. Gloomhaven: Jaws of the lion. *Gloomhaven Fandom Wiki*, 2022. URL https://gloomhaven.fandom.com/wiki/Gloomhaven:_Jaws_of_the_Lion.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition, 2025. URL https://web.stanford.edu/~jurafsky/slp3/. Online manuscript released January 12, 2025.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.emnlp-main.550.pdf.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, pages 9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Jiwei Li, Will Monroe, and Dan Jurafsky. A simple, fast diverse decoding algorithm for neural generation, 2016. URL https://arxiv.org/pdf/1611.08562.

Fábio Machuqueiro and João Piedade. Exploring the potential of modern board games to support computational thinking. In *2023 International Symposium on Computers in Education (SIIE)*, pages 1–8, 2023. URL https://www.researchgate.net/publication/376303721_EXPLORING_THE_POTENTIAL_OF_MODERN_BOARD_GAMES_TO_SUPPORT_COMPUTATIONAL_THINKING.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. Association for Computational Linguistics, 2019. URL https://aclanthology.org/D19-1410.pdf.

TruEra. Rag triad, 2024. URL https://www.truera.com/rag-triad-evaluation-framework/.

José P. Zagal. *Ludoliteracy: defining understanding and supporting games education*. ETC Press, 2011. URL https://www.academia.edu/23100280/Ludoliteracy_defining_understanding_and_supporting_games_education.