Daniel Williams
CSPB:4830
Prof. Curry Guinn
Spring 2025

Implementing a Basic RAG System for a Board Game

Abstract
In this project, I implement a basic RAG system for the board game Panzer Leader. I processed a PDF of the rulebook to extract each rule. I then created a tree representation of the rules. Each node in the tree was then vectorized and stored in a database for retrieval. During testing, each question in the test set was vectorized and then a cosine similarity score was calculated for each of the nodes. The three most similar nodes were retrieved and shown to the LLM as context for answering the question. Retrieval and generation were both evaluated. For retrieval, this simple similarity comparison resulted in correct retrieval 55% (22/40) of the time, partial retrieval 30% (12/40) of the time, and incorrect retrieval 15% (6/40) of the time. I tested four LLMs in the generation stage. The models performed similarly to each other (70-72.5% accuracy, or 28-29/40 questions answered correctly). Moreover, I found that when retrieval was successful, generation was almost guaranteed to be successful. Thus, improving retrieval is key to improving the system. This provides a benchmark for improving such RAG systems in future work.

Introduction

Retrieval-Augmented Generation (RAG) systems combine the strengths of information retrieval with the generative capabilities of large language models (LLMs), by retrieving relevant context to a query and inserting this context into the system prompt of an LLM. This has been shown to improve performance of question answering systems, and to be an effective way to leverage the QA abilities of LLMs over documents not included in the model's training data (Gao et al, 2023). My project implements and evaluates a RAG system over the board game Panzer Leader. I address the challenge of providing accurate answers to queries about rules that may arise during the course of game play.

The primary goals of this project are to:

1. Evaluate the effectiveness of vector similarity for rule retrieval from a structured game document
2. Compare the performance of different LLMs in generating accurate rule interpretations
3. Establish benchmarks for future improvements in RAG systems that have a structured knowledge base

Related Work

The fundamental concept of RAG involves expanding system prompts with relevant retrieved context. This helps reduce hallucination by "focusing" the LLM on key information. However, this introduces other challenges such as information integration and correctly identifying information gaps in the retrieved context. Several key studies have established the foundation for the approach implemented in this project.

Daniel Williams
CSPB:4830
Prof. Curry Guinn
Spring 2025

Benchmark research has identified four critical capabilities for effective RAG systems: noise robustness (ignoring irrelevant information), negative rejection (refraining from answering when context is insufficient), information integration (combining data from multiple sources), and counterfactual robustness (detecting mismatches between retrieved information and known facts) (Chen et al, 2024). These capabilities informed the discussion of retrieval and generation success.

There have been studies evaluating the effectiveness of RAG for documents that are similarly hierarchically structured. In technical documentation contexts, such as machine tool manuals that can span thousands of pages, researchers have employed prompt engineering alongside RAG techniques to reduce hallucination and improve answer accuracy (Cho et al, 2024). This approach parallels my project's focus on a complex rulebook, though without the fine-tuning component.

More sophisticated approaches for complex legal corpora have also emerged. Barron et al employed non-negative matrix factorization to extract latent topics and route queries to specialized vector stores, while also utilizing knowledge graphs to enhance retrieval and reasoning (2025). While this project employs a simpler approach with a single rulebook, these techniques suggest potential pathways for extending the system to accommodate larger game rule collections or multiple games.

Data

This project utilizes the rulebook for Panzer Leader, a hex-and-counter board wargame originally published by Avalon Hill in 1970. The primary data source is a facsimile of the official rules posted online, which was verified to match the original publication. The complete rulebook consists of 44 pages organized into 16 top-level sections covering various aspects of gameplay, such as movement, combat, and victory conditions.

The PDF rulebook was converted into a Python-amenable format using PyPDF. After extraction, regular expressions were employed to segment the text into a hierarchical tree structure. Each node in this tree corresponds to a specific rule section and contains three key components:

1. Title: Index to rule (e.g. XVI.A.1.a)
2. Text: The rule text
3. Relationships: Pointers to parent and child nodes, representing the hierarchical organization of the rulebook

This tree structure preserves the original organization of the rulebook while facilitating retrieval of specific rule sections.

For the embedding process, the BAAI/bge-large-en-v1.5 model was used to transform each tree node into a high-dimensional vector representation. This embedding model was selected for its strong performance in semantic search applications.

Daniel Williams
CSPB:4830
Prof. Curry Guinn
Spring 2025

To evaluate the RAG system, a test set of 40 questions was generated using Claude AI. This approach mirrors methodologies found in my background research (Chen et al, 2024), where LLMs are employed to create diverse and realistic test queries. The generation process involved passing sections of the rulebook to Claude and instructing it to create questions that players might naturally ask about those rules. The resulting test set spans all 16 sections of the rulebook. The generated answers were verified based my own knoweldge of the game to ensure accuracy.

Methodology

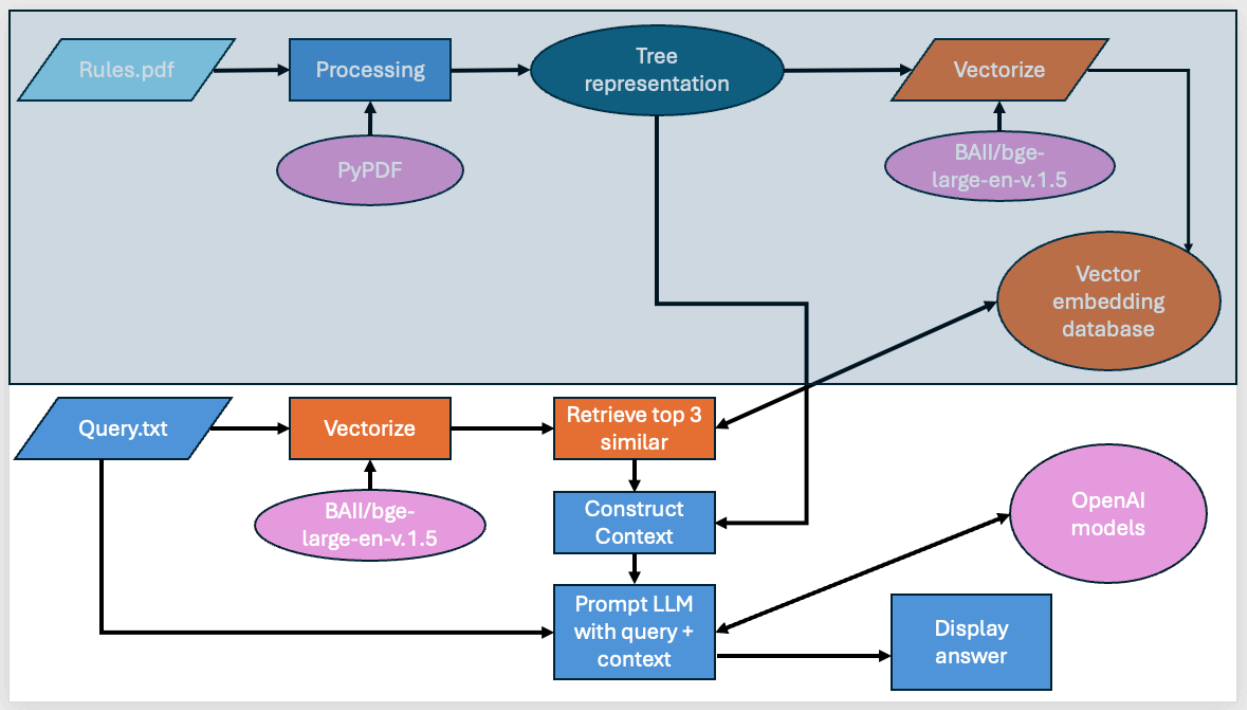

Figure 1

Figure 1 shows the flow of the RAG system. The top half of the figure, shaded in blue, shows the data processing/vectorization stage, and is described in the Data section. The bottom half incorporates two stages: retrieval and generation. In the retrieval stage, a test question is vectorized using the BAAI/bge-large-en-v.1.5 embedding model. Cosine similarity is calculated between this query vector and each node vector in the document tree. The three nodes with the highest similarity scores are selected. These nodes' IDs in the vector embedding database are then used to retrieve the appropriate text from the document tree, and this text is used as context in the system prompt for the LLM.

For generation, four different OpenAI LLMs were evaluated: 4o-mini, o3-mini, o1-mini, and 3.5-turbo. Each model received identical prompts containing the original question, the retrieved context, and instructions to answer the question based solely on the context.

Daniel Williams
CSPB:4830
Prof. Curry Guinn
Spring 2025

Evaluating the system required evaluating both retrieval and generation. For each question, the retrieval was classified as either *correct*, *partial*, or *incorrect*. Correct meant all information needed to answer the question was present in retrieved nodes. Partial meant some but not all required information was retrieved. Incorrect meant that the retrieved nodes did not contain information relevant to answering the question. For generation, success was determined by comparing model outputs against the ground truth answers. Answers were classified as "correct" or "incorrect".

My implementation used langchain libraries for accessing embedding models, performing similarity scores, and interacting with LLMs during the generation stage.

Results

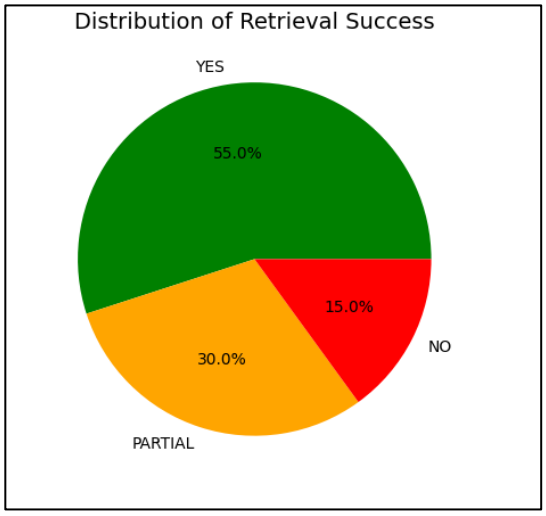

Figure 2

As shown in Figure 2, the retrieval component demonstrated modest success with a similarity-based approach. 55% of queries (22/40) resulted in correct retrieval; 30% (12/40) partial retrieval, and 15% incorrect retrieval. This indicates that while a basic vector similarity approach can work for a majority of cases, there is significant room for improvement.

Daniel Williams
CSPB:4830
Prof. Curry Guinn
Spring 2025

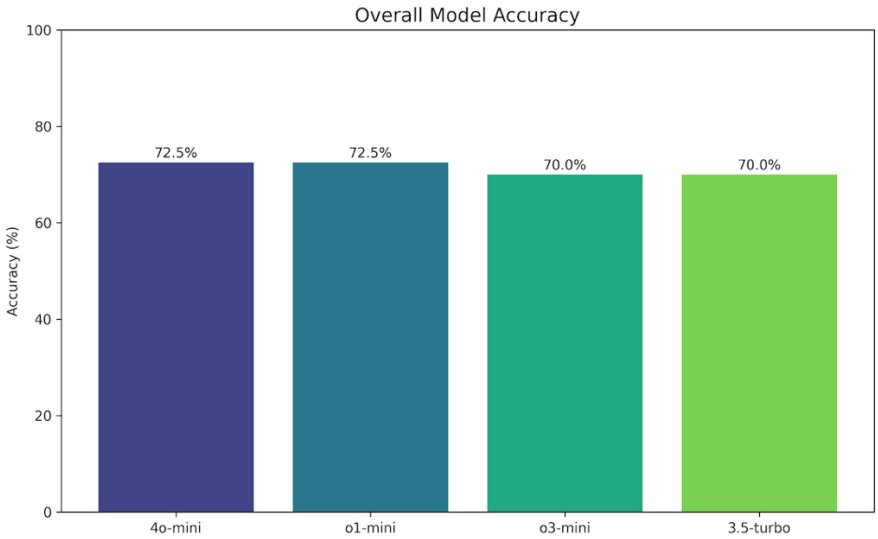

Figure 3

As shown in Figure 3, all four models achieved similar overall accuracy rates between 70-72.5% (28-29/40 questions answered correctly). This shows that model selection had minimal impact on the overall system performance.

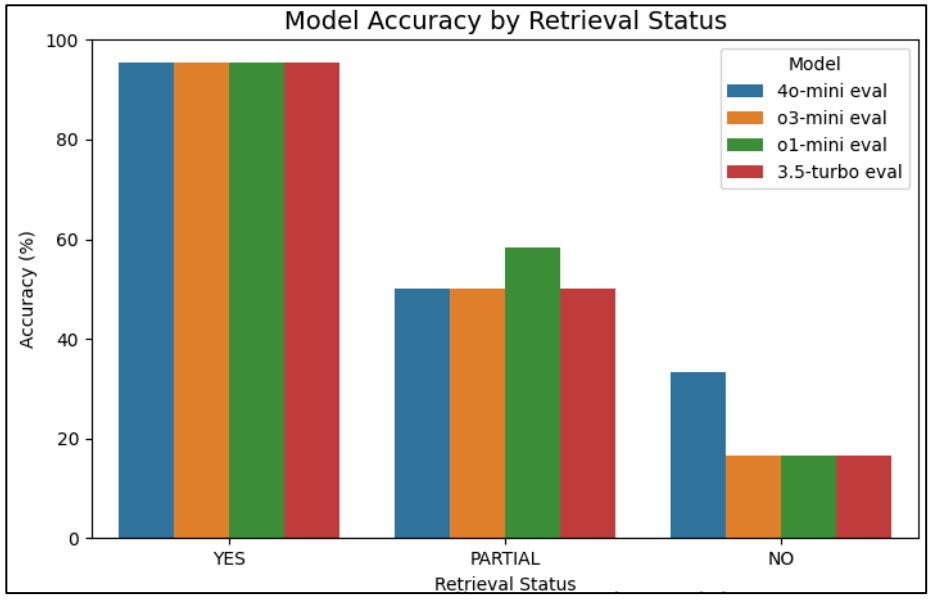

Figure 4

However, as shown in Figure 4, generation performance was strongly correlated with retrieval success. When retrieval was successful, all models achieved near perfect accuracy. With partial retrieval, accuracy dropped significantly to 50-60%. With incorrect retrieval, accuracy dropped

further still, to 15-35%. Notably, the 4o-mini model was able to outperform other models in cases of incorrect retrieval.

These results establish that retrieval is a critical bottleneck in the system. The near-perfect performance of all models with correct retrieval suggests that smaller, more efficient LLMs can be substituted in the generation stage, if retrieval is optimized.

Discussion

The results clearly point the way to improving retrieval in such systems. Interestingly, partial and incorrect retrieval still resulted in relatively high accuracy. With partial retrieval, it may be the case that some rules presuppose other rules, and thus the retrieved context contains the required information implicitly. For example, question 7 asks "Under what condition can a vehicular unit NOT enter a road hex at the road movement rate?" The best node to answer this question is rule V.D.3, which plainly states that the movement rate can only be used if there are no more than two units in the hex. However, rule V.D.5 contains a more specific scenario: "If entry into a particular hex is prohibited for a unit except by road movement, a unit may not enter that hex if there are MORE than two units on that hex." All models generalized from this specific scenario to produce the correct answer that if there are more than two units in a hex, the road movement rate cannot be used.

For cases of incorrect retrieval, it may be that the models are using their internal knowledge about board games to produce correct answers. This raises concerns about hallucination, however, as the models were instructed to answer questions based on the provided context. An example of negative rejection may illustrate the point. In question 39, "What is Panzerblitz Assault and how is it executed?", the retrieved context was incorrect – it did not refer to Panzerblitz Assault at all, but rather to various "CAT" (close-assault tactics) rules. 4o-mini correctly identified that the context did not answer the question – but then proceeded to speculate (incorrectly) about the answer before finally referring the user to refer to the rules for the precise answer. The other models, in contrast, simply attempted to answer the question (incorrectly) using the incorrect context.

Conclusion and Future Work

This project implemented a basic RAG system for the Panzer Leader board game rulebook, achieving 70-72.5% accuracy across four different LLMs. The most significant finding was the strong correlation between retrieval quality and generation accuracy, with all models achieving near-perfect performance when provided with correct context. This clearly identifies retrieval as the critical bottleneck in the system's performance, rather than the choice of generation model.

The hierarchical tree representation of the rulebook proved effective for organizing the content while preserving its structural relationships. The BAAI/bge-large-en-v1.5 embedding model demonstrated adequate performance in capturing semantic similarities between questions and

rule sections, though the 45% partial or incorrect retrieval rate indicates substantial room for improvement.

In terms of improving the retrieval side, Gao et al. (2023) identify several options for this, including query reformulation, where user questions are rephrased to better align with the database; hybrid retrieval, where vector similarity is combined with keyword-based methods; and re-ranking of initial retrieved results. For a hierarchically-structured document like a rulebook, expanding the context to include parent/child nodes when there is a similarity score between them can provide a more complete context, especially when a question requires combining information across multiple sections/subsections.

This implementation of a RAG system demonstrates both the promise and limitations of basic RAG. The fact that such a basic system can produce useful answers to a majority of queries is promising. However, much work can be done to improve such systems further.

Bibliography

Barron, R. C., Eren, M. E., Serafimova, O. M., Matuszek, C., & Alexandrov, B. S. (2025). Bridging Legal Knowledge and AI: Retrieval-Augmented Generation with Vector Stores, Knowledge Graphs, and Hierarchical Non-negative Matrix Factorization. *arXiv preprint arXiv:2502.20364*.

Chen, J., Lin, H., Han, X., & Sun, L. (2024, March). Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 16, pp. 17754-17762).

Cheng, X., Wang, X., Zhang, X., Ge, T., Chen, S. Q., Wei, F., ... & Zhao, D. (2024). xrag: Extreme context compression for retrieval-augmented generation with one token. *arXiv preprint arXiv:2405.13792*.

Cho, S., Park, J., & Um, J. (2024). Development of Fine-Tuned Retrieval Augmented Language Model specialized to manual books on machine tools. *IFAC-PapersOnLine*, *58*(19), 187-192.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, *2*.