Week 4 – RAG with LLM Assignment

Jack Lin 9/18/25

Introduction

This work investigates the use of Retrieval-Augmented Generation (RAG) pipelines to answer benchmark chess-related questions derived from our three research sources: Chess Bench with Stockfish, Maia-2, and ChessGPT. The goal was to assess how different retrieval settings and large language models (LLMs) handle reasoning tasks in a technically specialized domain.

Two categories of models were compared:

- **Track A:** Tiny Llama (Hugging Face)
- **Track B:** Gemini API (Google)

The retrieval process was varied along two parameters:

- **Embedding model:** e5-Small-V2 vs. MiniLM
- **Chunking sensitivity:** 300 chunks / 50 overlap vs. 500 chunks / 100 overlap.

Outputs were collected as CSV (Track A) and JSON (Track B) files for side-by-side evaluation.

Methodology

1. Candidate passages were retrieved from the chess research papers using dense embeddings. Smaller chunks (300/50) emphasized retrieval precision, while larger chunks (500/100) provided broader context.

2. TinyLlama produced answers using retrieved context. Gemini API generated responses under the same retrieval conditions.

3. Compare how embeddings and chunking influenced answer quality, coverage, and fluency across both model families.

Results

TinyLlama (Track A)

- Tiny Llama has a high sensitivity to retrieval quality. With e5-small embeddings and smaller chunks (300/50), answers were aligned but often incomplete.
- TinyLlama frequently defaulted to generic prompts that relied on quoting text, rather than synthesizing original responses when using MiniLM embeddings.
- Larger chunks (500/100) improved responses but sometimes was irrelevant information.
- e5-small consistently produced better grounding than MiniLM, which occasionally missed relevant passages.

Gemini API (Track B)

- Gemini produced coherent, literature aligned answers across both embeddings and chunking settings.
- It not only retrieved facts such as mentioning Stockfish annotations but also explained their implications for model training and evaluation.
- While chunking had minor influence, Gemini handled both 300/50 and 500/100 configurations smoothly and accurately.
- The responses showed Gemini's ability to reason about human-like play and policy–language integration which were two separate topics from different papers.

Comparative Insights

- Tiny Llama had limited synthesis and creativity while Gemini responded fluently and was able to create an organic response.
- Embedding quality influenced TinyLlama more than with Gemini. Both embedding models, MiniLM and e5-small, produced adequate results but TinyLlama responded more efficiently using e5-Small
- Tiny Llama is better for factual responses but only when using e5-small and smaller chunks.
- Gemini could reason or produce tasks and responded similarly under different embeddings and chunk sizes.

Conclusion

The experiments indicate that RAG pipelines are greatly influence by both embedding models and chunking sensitivity. TinyLlama was more computationally efficient but lacked reasoning abilities, especially with weaker embeddings (Mini LM) or larger chunking windows (500/100), thus reducing its quality in responses. The e5-small model combined with a smaller chunk size (300/50) benefited the response quality for Tiny Llama. Gemini provide robustness, adaptability, and deeper reasoning, making it well suited for specialized domains; however, it is computationally heavy. Gemini did not compromise the quality of its responses when using either embedding models or chunk sizes. Overall, for best results using RAG systems, pairing a high-quality embedding model with a capable generation model should achieve reliable performance.