Neurathink: Knowledge Graph Based Retrieval Augmented Generation

Varun Venkatadri, UNC Charlotte | Dr. Siddharth Krishnan, College of Computing and Informatics

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Introduction

Importance: RAG systems are the main way to give LLMs access to information they wouldn't have access to normally.

The problem: The most popular way of doing this is by using text as a context store. This makes it difficult for LLMs to understand the relationships between different entities, forcing it to infer them.

Our solution: Instead of using text as a context store, we can provide the LLM a subset of a knowledge graph. This will help the LLM understand relationships as they will be explicitly given to it.

Background

LLM: Large Language Model; an AI model that can be communicated with using natural language.

RAG: Retrieval Augmented Generation; a method of providing a prompt along with context to answer the prompt to an LLM.

Knowledge Graph (KGraph): A graph in which each node is some entity and each edge represents a relationship between them. Example: (Isaac Newton) — discovered → (Gravity).

Method

Developed 2 seperate RAG pipelines:

- Text-based RAG system that uses Wikipedia articles as the context store.
- KGraph based RAG system that uses GPT to encode Wikipedia articles into a knowledge graph that it uses as the context store.

Using the pipelines, conducted a series of 3 tests where an LLM was asked 20 questions about a topic:

Test 1

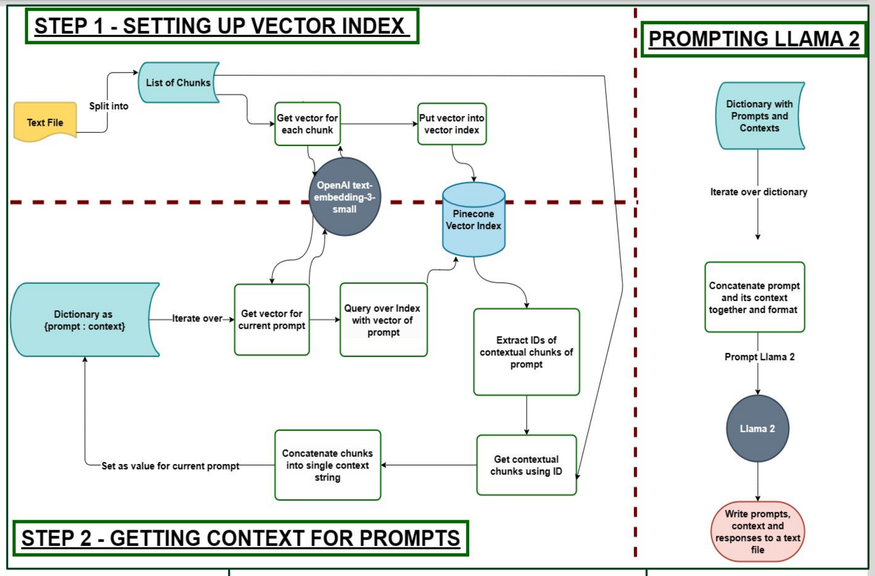
Text vs KGraph RAG in a scenario where the model already has some knowledge about the context. For this test, general history.

Test 2

Text vs KGraph RAG in a scenario where the model has no knowledge about the context. For this test, the 2023 Barbie movie.

Test 3

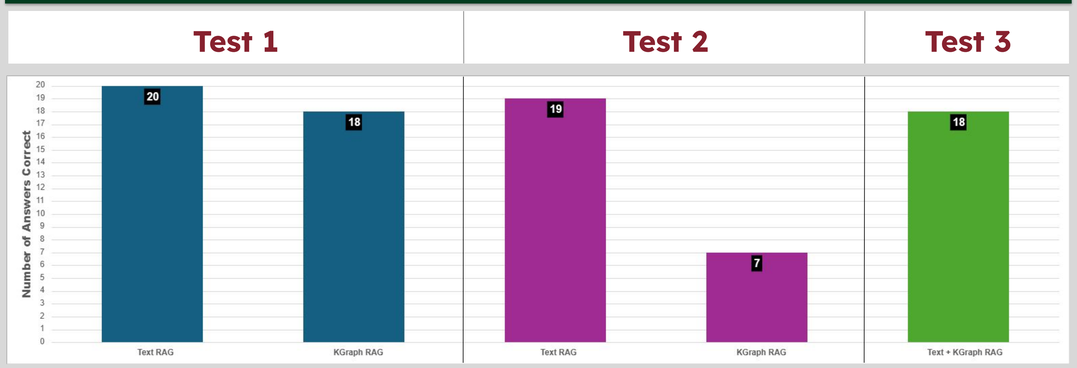
Text + KGraph RAG in the same scenario as test 2.



Text Based RAG Pipeline

Graph Encoding Models: GPT 3.5 T - 0125 (Test 1), GPT 4 T Preview (Test 2 & Test 3). RAG Models: LLama 2 (Test 1), GPT 3.5 T (Test 2 & Test 3)

Results



Future Plans

- What currently seems like the best way to move forward is to redesign the text-to-graph encoding process.
- Testing models like Gemini Ultra or Claude for this process may show better results.
- A possible solution to the graph encoding problem could be to split up the encoding process into 2 steps: node creation first, then relation creation using the nodes.

Conclusions

- Contrary to what we originally thought, Kgraph RAG actually performed significantly worse than regular text based RAG in all scenarios.
- This seems to be due to GPT's extremely poor ability to encode text to a KGraph. It tends to form repetitive nodes and loses lots of nuance while encoding.
- Overall, text RAG performed the best in tests 1 and 2. In test 3, text + KGraph RAG actually outperformed text RAG in terms of quality of answers, but overall got fewer answers correct.

References

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. IEEE Transactions on Knowledge and Data Engineering, https://doi.org/10.48550/arXiv.2306.08302.

Zou, X. (2020, March). A survey on application of knowledge graph. In Journal of Physics: Conference Series (Vol. 1487, No. 1, p. 012016). IOP Publishing, https://doi.org/10.1088/1742-6596/1487/1/012016.

Li, Y., Zhang, R., Liu, J., & Liu, G. (2024). An Enhanced Prompt-Based LLM Reasoning Scheme via Knowledge Graph-Integrated Collaboration. arXiv preprint arXiv:2402.04978, https://doi.org/10.48550/arXiv.2402.04978.