Members: 101, 136, 122, 104

Intro

With the recent developments in technologies such as machine learning and large language models (LLM's), there has been interest in using these new models for more specialized work and efforts have been made to specialize LLM's for specific businesses and areas of research. Currently, there are a few ways for people to utilize LLM's for their desired purposes, one recent development is RAG and the continuation, GraphRAG, which is what our assignment focuses on.

For our assignment, we were to use GraphRAG to improve question answering over a dataset of peace agreements. The assignment consists of several parts. Firstly, the dataset we used is PA-X which is a large data collection program focused on organized violence and weaponized conflict around the world. The goal was to get an LLM to parse this information from PDFs into a knowledge graph format. Then, this information was put into a graph database and used for retrieval by an LLM to answer questions more accurately.

Background

The core concept of this assignment was to create a GraphRAG for our specified dataset and explore how this concept works in practice. A GraphRAG is based on RAG which stands for Retrieval-Augmented Generation. RAG is a technique used for LLM's so that they can work in domain specific information, such as internal company data. However, RAG has limitations when it comes to more nuanced and complex connections.

GraphRAG is a version of RAG that utilizes graph structure and the advantages that come with the relational structure of graph databases. GraphRAG can use the graph structure as an ontology or a knowledge graph. The benefit of this is that knowledge graphs support reasoning over entities and their relation. GraphRAG performs better than normal RAG, for example, in Mosolygo et al. (2024), when creating answers from multiple sources or intricate relationships between entities for news reporters.

System Architecture and Ingestion Pipeline

The process of creating the GraphRAG firstly involves scraping peaceagreements.org, then downloading and processing the raw text using selenium. The dataset we chose consists of 2000+ PDFs of peace agreements or accords. The main technique for processing this amount of data for an LLM, is to create chunks for the knowledge graph. Our chosen chunk size was 1500 with an overlap of 200. The reason we went for a larger chunk size was that the documents might present more complex information. Bhat et al. (2025) investigates

chunking techniques and a part of their research involves chunk sizing. They argue that in their findings, larger chunking allows for wider retrieval windows which is beneficial for data where context is important. Since our data is context dependent, we decided to use a larger chunking size. OCR was also used to parse text, since some pdfs are not actual text, but images instead.

After the data was preprocessed, the next step was to create the ontology and the schema. LangChain was our preferred library, compared to other libraries such as LlamaIndex, as we get useful metadata about the schema generated and LangChain is more easily integrated into the databases. LangChain also offers built-in vector support, something that was essential for our specific use case, as we rely on a Neo4j database.

Then from our yaml file, we extract the schema and create the knowledge graph. The knowledge graph is created by an LLM, so we prompt it to extract the nodes and relationships according to the schema rules we have. We then have our completed knowledge graph following the rules of the ontology we created. Then, with our knowledge graph, we start the graph database construction. For our project we used Cypher and Neo4j. We deliberated on using SPARQL as SQL was already familiar to some of us and had used SPARQL before. However, we chose Cypher as Neo4j is better at handling more complex data and relationships compared to RDF based databases and SPARQL, as stated in Zhao et al. (2023), where the researchers try to reconcile the two.

Knowledge Graph Schema

In order to properly organize the information we extract from the peace agreements we needed to design a schema which would serve as our blueprint for how the information would be structured in the knowledge graph. The schema formally defined which entities, relationships and constraints that would be permitted in the resulting graph. This kind of explicit definition becomes crucial when transforming unstructured text, such as the peace agreements, into a structured and queryable graph (Scaffidi et al., 2025).

The core of our schema is the defined entities and relationships, and the design choices we made depended on central and important themes in the peace agreements, and what questions the graph was expected to be able to answer. The defined entities represent the objects and concepts found in the peace agreements, such as “Agreement”, “Party”, “Topic” and “Location”. We chose to define 18 entities to properly capture the more detailed components of the peace agreements, from the central entity “Agreement” to the more intricate entities “MonitoringBody” and “Provision”. We also defined multiple relationships

which serve to explicitly model the interactions and obligations in the agreements. An example of this is the structural constraint “OBLIGATES: Agreement | Clause -> Party”, that enables the LLM to query which party is bound by which clause in the peace agreement.

A point to discuss is the GraphRAG with and without a proper schema. We first created a graph database without a proper schema and let the LLM choose more freely its own rules and ontology. What we saw happen was that there were certain language differences for entities that the schemaless LLM did not catch. A practical example was when querying for the United Nations, it was not always connected in the way we expected. When investigating, we saw that the United Nations often had different classifications. Some had organization and others had organisation. This is obviously caused by the difference in spelling for British English and American English.

Retrieval system

The GraphRAG utilizes three retrieval methods. Firstly, the retrieval system creates a question embedding using an OpenAI embedding model, and queries the Neo4j vector index for the top 5 chunks with the closest question embedding vector. Then it does a full text search by matching the prompt’s keywords and the chunked documents and retrieves the top 5 most relevant results. Lastly, it does a knowledge graph search. It performs a graph pattern match in Neo4j and finds the entities where the names contain the query string or parts of it and retrieves the documents from the found entities. Optionally, after the three methods, it fuses all the results using reciprocal rank fusion which helps improve the retrievals accuracy. The top results are then concatenated with a character limit into a context string. Lastly the context and question is sent to the LLM via an API and creates the final answer.

Literature Review

There is an ongoing rapid development of RAG systems with variations focused on meeting requirements for their specific purpose and use-case. Traditional RAG systems, often called VectorRAG, excel at handling a large volume of data and answering queries that can be answered based on localized data segments. However, a limitation arises when there is a need for answering sensemaking queries that require an understanding of an entire dataset. In the research paper “From Local to Global: A GraphRAG Approach to Query-Focused Summarization” Edge et al. (2025), a GraphRAG framework is proposed which builds on traditional RAG, but involves knowledge graphs to enable sensemaking. First, an LLM is used to build a knowledge graph from the source documents, deriving the entities and

relationships. Then it separates the graph into communities, and generates community summaries for all the groups. Next, when the LLM is given a query, it is answered using a map-reduce process on the summaries, which then generates the final global answer. Edge et al. (2025) found that GraphRAG compared to classic RAG has substantial improvements in both comprehensiveness and the diversity of generated answers, thus enabling it to properly answer sensemaking queries over an entire large text corpus (Edge et al., 2025).

However, GraphRAG has a set of limitations, as addressed by Papageorgiou et al. (2025) in their article “Hybrid Multi-Agent GraphRAG for E-Government: Towards a Trustworthy AI Assistant”. They introduce a hybrid, multi-agent version of GraphRAG that is created to meet the challenges of e-government applications. AI-driven virtual assistants are increasingly being adopted by public institutions, and it is essential that they deliver comprehensive, credible and factually accurate question answering. Their hybrid GraphRAG was evaluated using the European Commission’s Press Corner as a data source to test their multi-agent architecture. To optimize the knowledge discovery they developed a framework that combines traditional RAG, embedding-based retrieval, real-time web search, and structured graphs generated by LLMs. They highlight the importance of reducing LLM hallucinations, reinforcing the factual grounding, and enhancing the quality of more complex responses. If the public-sector AI applications fail to meet these requirements it could lead to serious consequences, such as a government being responsible for spreading misinformation and losing trust with the public (Papageorgiou et al., 2025).

In the research paper “Medical GraphRAG: Towards Safe Medical Large Language Model via Graph Retrieval-Augmented Generation”, Wu et al. (2024) also propose a variation of GraphRAG. They address the limitations of GraphRAG in the medical field, specifically noting the lack of evidence-based response generation and credibility. In a specialized field such as medicine it is extremely important that the LLMs do not modify, hallucinate or introduce creative elements into the data it generates. Fitting the medical knowledge that has been accumulated over thousands of years into current LLMs is hopeless, because of the finite context window. Further, in the field of medicine, precise terminology and established truths are essential, and must not be distorted or modified by an LLM. Thus, Wu et al. (2024) developed an enhanced version of GraphRAG, called MedGraphRAG, where they keep the complex reasoning ability of GraphRAG systems, but improve the graph construction to ensure that the responses are authenticated. MedGraphRAG uses Triple Graph Construction to link the data to credible sources and proper vocabularies, and U-Retrieval to help balance the global context awareness with precise indexing, which improves the response quality with few costs. From the results Wu et al. (2024) explains that

MedGraphRAG ensures that responses include source documentation and definitions of terminology, and that it consistently outperforms state-of-the-art models across all bench-marks.

Evaluation

To fact check the GraphRAG retrieval, the Neo4j database is structured so that every chunk generated by the LLM from the pdfs contains the meta data, including the pdf it pulled from. Therefore, every agreement and its relationship will have the original pdfs name, location, embedding vector etc. Another way to guard against hallucination is to ask the GraphRAG the same question multiple times to check for discrepancies in the answers from the LLM. We noticed few differences in the prompts we asked, and the GraphRAG generally stayed on topic with minimal hallucinations. We also used DeepEval to automatically check retrieval accuracy and amount of hallucination. DeepEval was chosen as it is a good tool for RAG analysis and evaluation as well as working well with OpenAI (Awan, 2025). DeepEval has the ability to check relevance in the GraphRAGs answer, which is a metric GraphRAG specifically works toward improving. This, along with the ability to analyse several steps of the GraphRAG process, made DeepEval the chosen automatic analysis tool for this assignment.

Conclusion

The GraphRAG had good adherence to the specified dataset, often referring to or mentioning the documents it used. This signals that it's actively using and interacting with the Neo4j database. It also handled prompted questions that were wrong and didn't fill the knowledge gap with hallucinations. We gave it a prompt about an agreement that was non-existent, and the model explicitly states that the retrieval produced no matching documents for the prompt. This again shows that the model is working with the chosen dataset. The model did however, sometimes struggle with classification issues, for example in the prompt about the Libya DDR declaration, it stated it was a peace agreement, instead of a declaration. It is a soft hallucination, but not a significant one. Also, when prompted, it gave more generalized answers, since the dataset is so vast, that it sometimes lacked precision. This could be solved by parsing the pdfs better, setting a limit on how many documents per conflict or agreement and filtering out the most relevant ones. In general however, we felt the model performed well and utilized the GraphRAG system efficiently.

Sources

Awan, A. (2025, January 14). *Evaluate LLMs Effectively Using DeepEval: A Practical Guide*. Datacamp. Retrieved December 3, 2025, from https://www.datacamp.com/tutorial/deepeval

Bhat, S. R., Rudat, M., Spiekermann, J., & Flores-Herr, N. (2025). Rethinking chunk size for Long-Document Retrieval: A Multi-Dataset analysis. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2505.21700

Zhao, Z., Ge, X., Shen, Z., Hu, C., & Wang, H. (2023). S2CTrans: Building a Bridge from SPARQL to Cypher. In *Lecture notes in computer science* (pp. 424–430). https://doi.org/10.1007/978-3-031-39847-6_33

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., & Larson, J. (2024, April 24). *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*. arXiv.org. https://arxiv.org/abs/2404.16130

Mosolygo, B., Fatemi, B., Rabbi, F., & Opdahl, A. (2024). Evaluating GraphRAG's role in improving contextual understanding of news in newsrooms. *Norsk IKT-konferanse for Forskning Og Utdanning*, *1*. https://doi.org/10.5324/nikt.6231

Papageorgiou, G., Sarlis, V., Maragoudakis, M., & Tjortjis, C. (2025). Hybrid Multi-Agent GraphRAG for E-Government: towards a trustworthy AI Assistant. *Applied Sciences*, *15*(11), 6315. https://doi.org/10.3390/app15116315

Scaffidi, H., Hodkiewicz, M., Woods, C., & Roocke, N. (2025). GraphRAG on Technical Documents - Impact of Knowledge Graph Schema. *Dagstuhl Research Online Publication Server*. https://doi.org/10.4230/tgdk.3.2.3

Wu, J., Zhu, J., & Qi, Y. (2024). Medical Graph RAG: towards safe medical large Language model via Graph Retrieval-Augmented Generation. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2408.04187

Appendix

Flow Chart of Solution

A visualization of our solution using a flow chart.