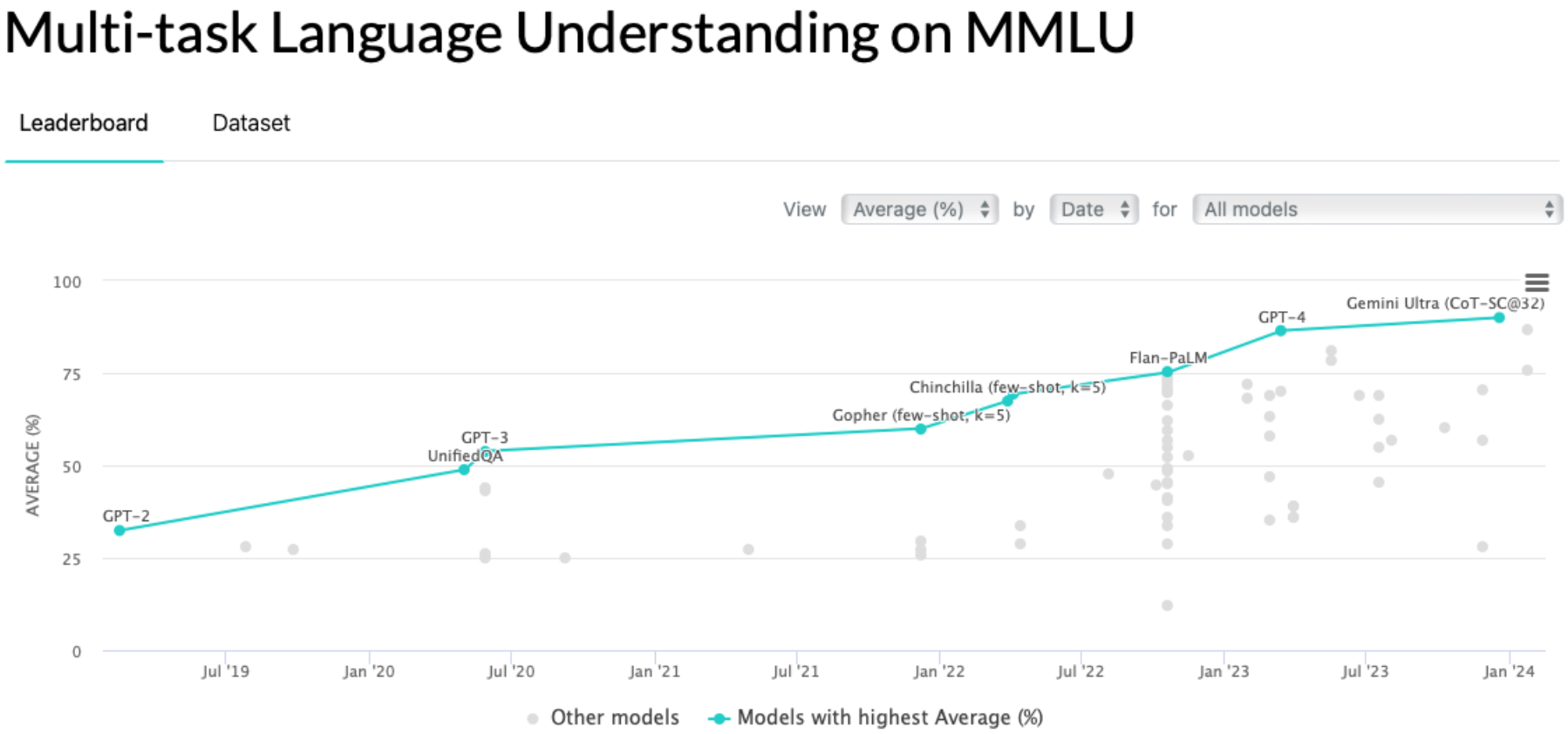
Retrieval Augmented Generation with Knowledge Graphs

JongGeun Lee

Overview

1. Introduction
2. What is RAG?
3. What to Retrieve?
4. Knowledge Graphs With LLMs
5. Conclusion

1. Introduction – Performance of LLMs



Large Language Models achieved a score higher

than the human expert level of 89.3% on the MMLU benchmark

- PapersWithCode, "Multi-task Language Understanding on MMLU", (2024).

1. Introduction – Drawbacks

***Drawbacks of LLM**

- ✓ Hallucination
- ✓ Outdated information
- ✓ Lack of in-depth knowledge in specialized domains

***Practical Requirements of Application**

- ✓ Domain-specific accurate answering
- ✓ Frequent updates of data
- ✓ Traceability and explainability of generated content
- ✓ Controllable Cost
- ✓ Privacy protection of data

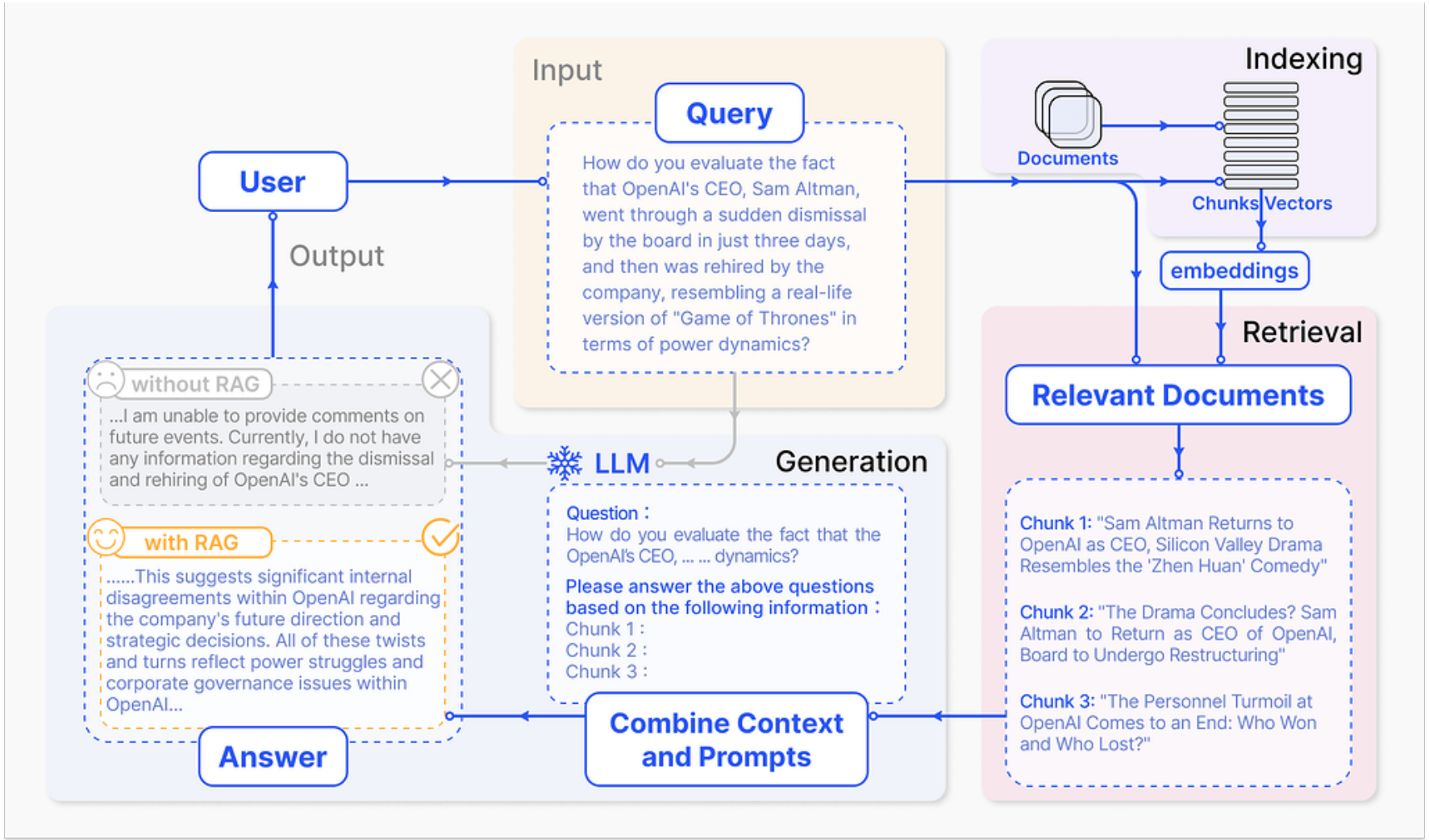
→ **Retrieval Augmented Generation**

2. What is RAG?

When answering questions or generating text, it first retrieves relevant information from a large number of documents, and then LLMs generates answers based on this information.

By attaching a external knowledge base, there is no need to retrain the entire large model for each specific task.

The RAG model is especially suitable for knowledge-intensive tasks.



- Gao et al. "Retrieval-Augmented Generation for Large Language Modles: A Survey". (2023).

3. What to retrieve ?

coarse

Retrieval granularity

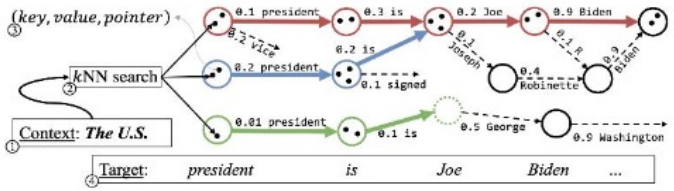
meticulous

low

level of structuration

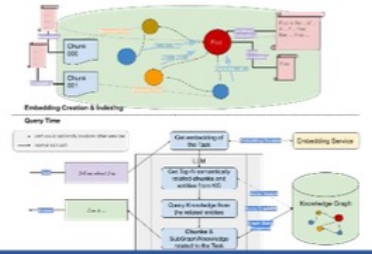
High

Chunk | In-Context RAG 2023



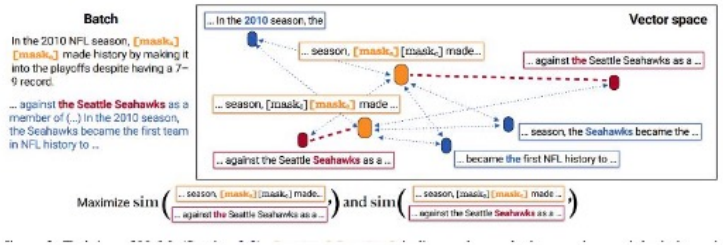
The search is broad, recalling a large amount of information, but with low accuracy, high coverage but includes much redundant information.

Knowledge Graph | 2023

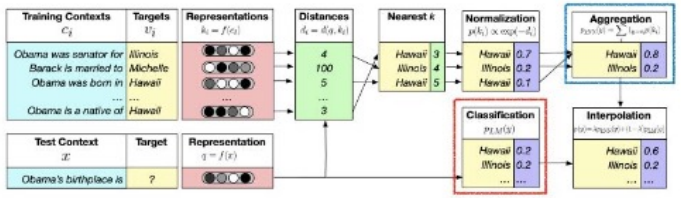


Richer semantic and structured information, but the retrieval efficiency is lower and is limited by the quality of KG.

Phrase | NPM 2023

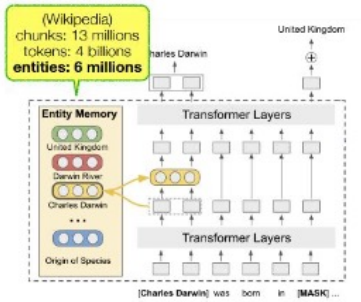


Token | KNN-LMM 2019



It excels in handling long-tail and cross-domain issues with high computational efficiency, but it requires significant storage.

Entity | EasE 2022



- Gao et al. "Retrieval-Augmented Generation for Large Language Modles: A Survey". (2023).

4. Knowledge Graphs With LLMs

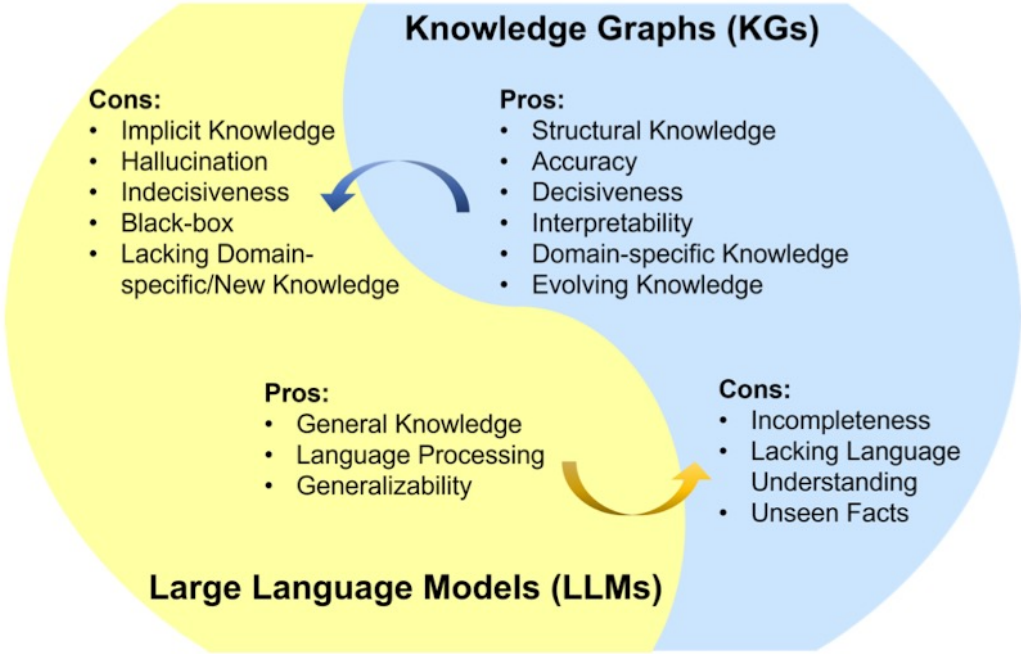


Fig. 1. Summarization of the pros and cons for LLMs and KGs. LLM pros: *General Knowledge* [11], *Language Processing* [12], *Generalizability* [13]; LLM cons: *Implicit Knowledge* [14], *Hallucination* [15], *Indecisiveness* [16], *Black-box* [17], *Lacking Domain-specific/New Knowledge* [18]. KG pros: *Structural Knowledge* [19], *Accuracy* [20], *Decisiveness* [21], *Interpretability* [22], *Domain-specific Knowledge* [23], *Evolving Knowledge* [24]; KG cons: *Incompleteness* [25], *Lacking Language Understanding* [26], *Unseen Facts* [27]. Pros. and Cons. are selected based on their representativeness. Detailed discussion can be found in *Appendix A*.

- Pan et al. "Unifying Large Language Models and Knowledge Graphs: A Roadmap". (2024).

4. Knowledge Graphs With LLMs

*GNP (Tian et al., 2023)

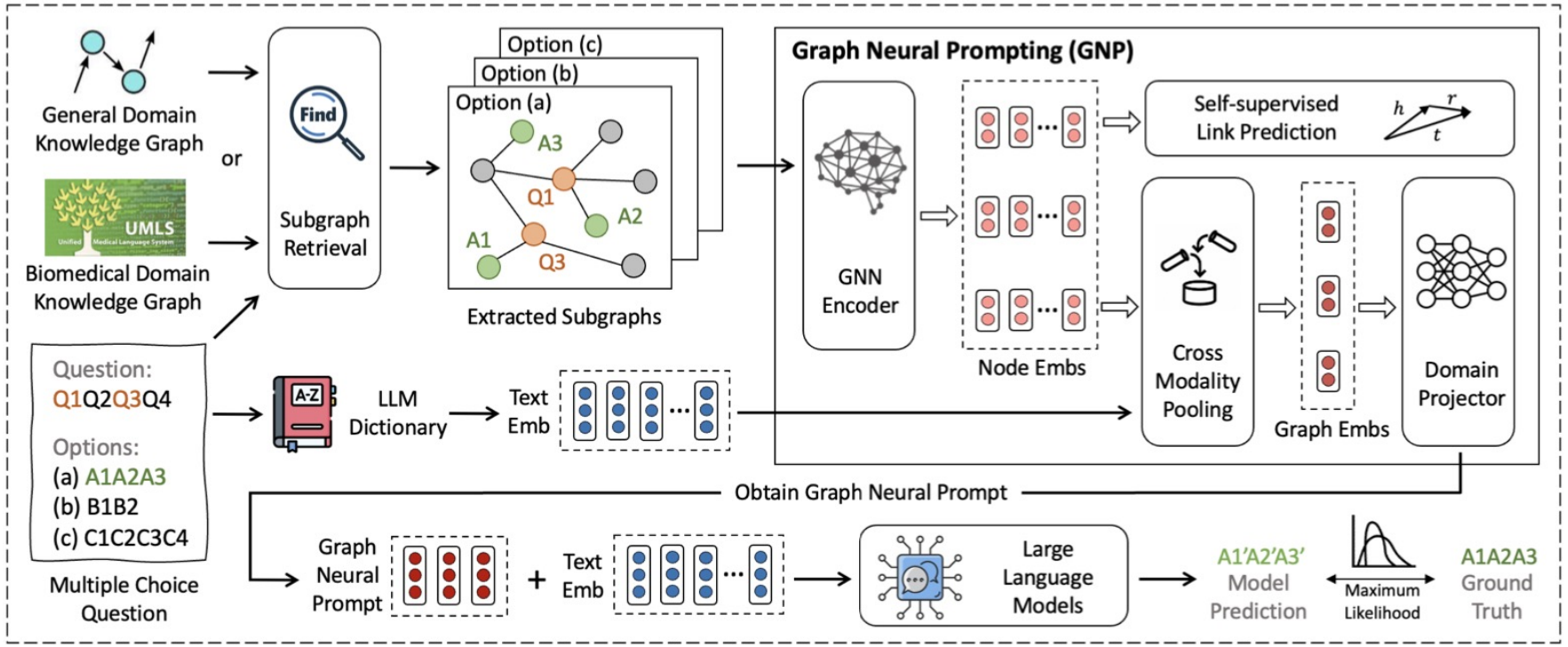


Figure 2: The overall framework. Given a multiple choice question, we first retrieve subgraphs from the knowledge graph based on the entities in the question and options. We then develop Graph Neural Prompting (GNP) to encode the pertinent factual knowledge and structural information to obtain the Graph Neural Prompt. GNP contains various designs including a GNN, a cross-modality pooling module, a domain projector, and a self-supervised link prediction objective. Later, the obtained Graph Neural Prompt is sent into LLM for inference along with the input text embedding. We utilize the standard maximum likelihood objective for downstream task adaptation, while LLM is kept frozen or tuned depending on different experimental settings.

- Tian et al. "Graph Neural Prompting with Large Language Models". (2023).

4. Knowledge Graphs With LLMs

*SURGE (Kang et al., 2023)

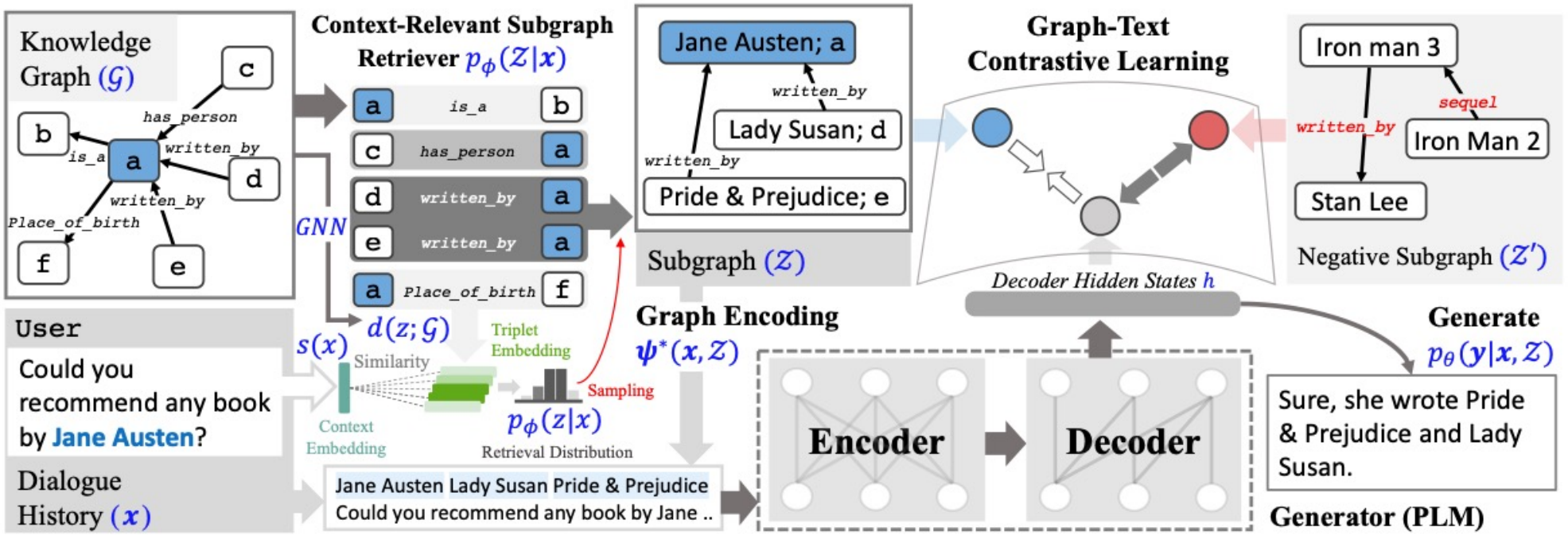


Figure 2: **Framework Overview.** Our framework, SURGE, consists of three parts. First, a context-relevant subgraph retriever $p_\phi(\mathcal{Z}|\boldsymbol{x})$ retrieves the subgraph $\mathcal{Z}$ relevant to the given dialogue history $\boldsymbol{x}$ from a knowledge graph $\mathcal{G}$ (e.g., 1-hop KG from entity *Jane Austen*; a). Specifically, we measure the similarity of a context and triplet embedding to compose the retrieval distribution $p_\phi(z|\boldsymbol{x})$ (§ 3.3). Then, we encode the retrieved subgraph $\mathcal{Z}$ using the graph encoding $\psi(\boldsymbol{x},\mathcal{Z})$ (§ 3.4). Finally, we use contrastive learning to enforce the model to generate a knowledge-grounded response with the retrieved subgraph (§ 3.5).

- Kang et al. "Knowledge Graph-Augmented Language Models for Knowledge-Grounded Dialogue Generation". (2023).

5. Conclusion

***Further Research Topics**

- ✓ Methods for aligning text and graph structure during the pretraining stage
- ✓ Multi-Modality: In addition to aligning graphs and text, research on further aligning images
- ✓ How many hops or how many subgraphs should be retrieved?
- ✓ What kind of domain projector would be appropriate ?
- ✓ What form would be most effective when soft-prompting graphs?

Thank You