FRI Natural language processing course 2025
Project report

University *of Ljubljana*
Faculty *of Computer and Information Science*

Integrating Structured Knowledge into Large Language Models

Evgenii Posashkov, Biljana Vitanova, and Žan Štuhec

Abstract

Large Language Models (LLMs) have demonstrated significant success in natural language processing tasks but still face challenges, one of which is handling complex question answering. This project explores methods to integrate structured knowledge, in form of knowledge graphs (KGs), into LLMs to enhance their performance. We aim to evaluate different techniques for integrating KGs. The ultimate goal is to assess whether the inclusion of structured knowledge can address the existing limitations of LLMs, and improve their reasoning capabilities.

Keywords

Largle Language Models, Knowledge Graphs, Question Answering, Reasoning

Advisors: Slavko Žitnik

Introduction

Despite advances in text generation by LLMs, they struggle with complex reasoning and fact recall. One approach to overcoming these limitations is integrating Knowledge Graphs (KGs), which provide structured representations of entities and relationships.

Our project explores methods to incorporate KGs into LLMs by pairing questions with relevant KGs, bridging the gap between unstructured text and structured knowledge, enabling LLMs to access and utilize detailed information for advanced reasoning and accurate conclusions.

By using both textual and structural elements of KGs, we aim to improve the accuracy of LLMs in complex question-answering scenarios.

Related Work

Recent research has explored multiple ways to integrate KGs into LLMs to better reasoning and improve task performance. Tang et al. (2024) introduced GraphGPT [1], which pairs LLMs with graph structures, aligning closely with our goal of improving question answering through KG integration. Kau et al. (2024) examined how KGs can mitigate issues like hallucinations in LLMs [2], which is relevant to our focus on improving factual accuracy. Yao et al. (2019) developed KG-BERT for knowledge graph completion [3], inspiring our use of graph embeddings in LLMs for better performance in complex reasoning tasks. Yasunaga et al. (2022) proposed DRAGON, a model that fuses text and KG during pretraining [4], which relates to our aim of combining textual and structural elements of KGs to enhance LLM capabilities. Kang et al. (2023) presented SURGE for knowledge-grounded dialogue [5], which aligns with our goal of ensuring that KGs improve LLMs factual accuracy in more complex tasks. And in the end Wang et al. (2017) surveyed KG embedding techniques [6], providing the basis for our exploration of graph embeddings like TransE and node2vec to better LLM performance. These studies together inform our methodology for integrating KGs into LLMs to improve their reasoning and accuracy in complex question answering.

Representation and Integration of KGs

Knowledge graphs (KGs) are constructed and integrated into the question-answering (QA) pipeline to improve the performance of large language models (LLMs) on complex factual questions. This section outlines the methodology for representing and utilizing knowledge graphs based on structured data retrieved from Wikidata.

To represent structured information, we use subgraphs extracted from Wikidata via SPARQL queries. Each subgraph is centered around a specific entity (e.g., a head of state) and captures a set of semantically relevant property-value pairs while omitting overly general or trivial attributes.

The subgraphs are modeled as directed graphs using the NetworkX Python library, where nodes represent entities such as leaders, countries, awards, or affiliations. Relationships

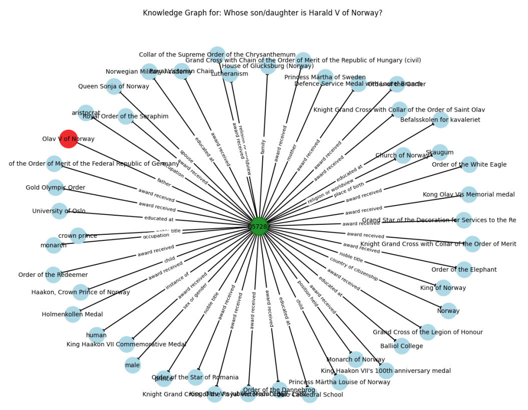
Knowledge Graph for: Whose son/daughter is Harald V of Norway?
Collar of the Supreme Order of the Chrysanthemum
Queen Sonja of Norway
Royal Order of the Seraphim
Olav V of Norway
Gold Olympic Order
University of Oslo
crown prince
monarch
Order of the Redeemer
Haakon, Crown Prince of Norway
Holmenkollen Medal
human
King Haakon VII Commemorative Medal
male
Order of the Star of Romania
Order of the White Eagle
Kong Olav V's Memorial medal
Order of the Elephant
King of Norway
Norway
Grand Cross of the Legion of Honour
Balliol College
Monarch of Norway
King Haakon VII's 100th anniversary medal
Princess Märtha Louise of Norway
Church of Norway
Skaugum
Befalsskolen for kavaleriet
Knight Grand Cross with Collar of the Order of Saint Olav
Lutheranism
House of Glücksburg (Norway)
father
child
award received
educated at
occupation

Figure 1. Knowledge Graph for the query: "Whose son/daughter is Harald V of Norway?". The green central node represents Harald V (Q5728), and blue peripheral nodes represent related entities and values. Edges are labeled with property types such as: father, child, award received, and educated at.

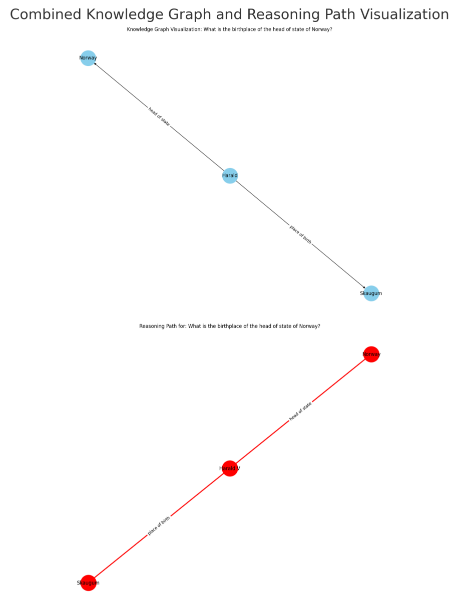
Combined Knowledge Graph and Reasoning Path Visualization
Knowledge Graph Visualization: What is the birthplace of the head of state of Norway?
Norway
head of state
Harald
place of birth
Skaugum
Reasoning Path for: What is the birthplace of the head of state of Norway?
Norway
head of state
Harald
place of birth
Skaugum

Figure 2. Combined Knowledge Graph and Reasoning Path Visualization for the question: "What is the birthplace of the head of state of Norway?"

between these entities are encoded as labeled edges. When available, human-readable labels are assigned to nodes to improve interpretability in downstream tasks.

For each leader entity, we extract up to 20 related RDF triples using SPARQL. To focus on meaningful information, we filter out low-value properties such as gender or generic classifications like "instance of human." The remaining data is then formatted both as a list of triples and as a compact graph representation.

The current implementation focuses on preparing data for structured knowledge integration. For each question-answer pair, the corresponding knowledge graph subgraph is extracted, visualized, and stored in both serialized triple format and as graph objects. These subgraphs form the basis for retrieval-based evaluation and further downstream QA analysis.

Figure 2 illustrates a RAG setup using a knowledge graph. The top part shows the original graph structure, including triples like (Harald, head of state, Norway) and (Harald, place of birth, Skaugum). These are converted into natural language sentences (e.g., Harald is the head of state of Norway.) and stored in a vector index for retrieval. The lower part visualizes a possible reasoning path the system might follow to answer a query first identifying that Harald is the head of state of Norway, then linking to his place of birth to infer the final answer, Skaugum. This highlights how structured triples support fact retrieval and multi-step reasoning in RAG pipelines.

Methodology

We aim to build a RAG system where knowledge graph triplets (e.g., (Slovenia, president, Nataša Pirc Musar)) are first converted into human-readable textual chunks.

One such example is:

> (Slovenia, president, Nataša Pirc Musar)→ "The president of Slovenia is Nataša Pirc Musar."

These are embedded and stored in a vector database. When a user asks a question, the system retrieves the most relevant statements based on semantic similarity and provides them to a language model. The model then uses this context to generate an answer.

We then compare the answer quality across different representations of knowledge graph data. Specifically, we test how the use of raw triples versus human-readable sentences affects retrieval and the final LLM response.

LLM frameworks

We initially tested three smaller language models: Phi-1.5, DeepSeek Coder, and TinyLlama, each with approximately 1 to 1.5 billion parameters. The goal was to evaluate how well these models handle questions about world leaders when provided with structured contextual information. Among the three, Phi-1.5 consistently produced the most accurate answers. In contrast, DeepSeek Coder often failed to provide correct responses, which is expected given its focus on code generation rather than question answering. Based on this, we will further focus on using larger LLMs that are better suited for QA tasks.

Evaluation Dataset

In our opnion Wikidata is the best choice for integrating structured knowledge into large language models because it provides rich and fine-grained relationships between entities, offering much more than basic facts and enabling complex, multi-hop question answering tasks. Unlike DBPedia, which is extracted from Wikipedia and sometimes loses structure, Wikidata was designed from the ground up as a structured knowledge graph, making it highly standardised and easy to work with. Its public SPARQL endpoint allows for easy and powerful querying to extract relevant subgraphs aligned with specific QA tasks. Wikidata is continuously updated by a global community, ensuring that the information remains accurate and up-to-date, which is critical when evaluating modern LLMs. It also provides rich metadata such as multilingual labels, descriptions, aliases and qualifiers, which can further enhance the quality of retrieval and generation.

Wikidata is used as the main source of structured knowledge in this project. We extract a domain-specific subset focused on world leaders and their related facts. The evaluation is based on a curated set of 20–50 question-answer pairs that require multi-hop reasoning over these subgraphs. Using Wikidata offers a large and reliable knowledge base for testing how well different methods integrate structured data into language models.

Experiments

When prompting, we observed that the model's accuracy is highly sensitive to how structured knowledge is encoded and retrieved. Using vector similarity search, the model attempts to answer the questions. However, if entities are represented as identifiers (Wikidata QIDs), the model often misinterprets the context. One such example is shown in Table 1.

Table 1. Effect of context representation on the model's answer to the question: Who is the head of state of Italy? The correct answer is Sergio Mattarella.

Input Type	Retrieved Entity Label	Model Answer
No context	–	Carlo
Without label mapping	Q3956186	Bernardo Mattarella
With label mapping	Sergio Mattarella	Sergio Mattarella

Discussion

While the baseline relies on standard embedding-based retrieval, we also explored potential approaches for improving LLM performance through training-time integration of structured knowledge. These include techniques such as:

- Graph embedding algorithms (e.g., TransE, Laplacian Eigenmaps, node2vec, DeepWalk)
- Attention-based fusion of KG facts with textual input via transformer architectures

However, due to the complexity of these methods, we currently focus on direct injection of structured knowledge into the prompt and aim to iteratively improve the effectiveness of this approach.

References

[1] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 491–500, 2024.

[2] Amanda Kau, Xuzeng He, Aishwarya Nambissan, Aland Astudillo, Hui Yin, and Amir Aryani. Combining knowledge graphs and large language models. *arXiv preprint arXiv:2407.06564*, 2024.

[3] Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

[4] Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy S Liang, and Jure Leskovec. Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35:37309–37323, 2022.

[5] Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. Knowledge graph-augmented language models for knowledge-grounded dialogue generation. *arXiv preprint arXiv:2305.18846*, 2023.

[6] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE transactions on knowledge and data engineering*, 29(12):2724–2743, 2017.