Knowledge Graphs and Retrieval-Augmented Generation (RAG)

April 15, 2025

1 Introduction to Knowledge Graphs

A **knowledge graph** represents a network of entities (objects, events, situations, or concepts) and illustrates the relationship between them. It is typically stored in a graph database and visualized as a graph structure.

- **Node:** person/place/thing
- **Edge:** relationship between the nodes

Utility:

- Discern the meaning of homographs (same spelling, different meaning)
- Understand hidden underlying connections between nouns to process context

A knowledge graph is a structured representation of text, often stored in the format:

```
(Subject, Predicate, Object)
```

This format captures relationships between two entities.

2 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is a technique used to enhance the accuracy of generative models using external data sources.

Without RAG

The large language model (LLM) takes user input and generates a response based only on its training data.

With RAG

a) The user input is used to retrieve information from an external data source.

b) The query and the retrieved context are both fed into the LLM.

c) The LLM combines the new knowledge with its internal training to generate more accurate responses.

Components of RAG

- **Create External Data:** Data can come from APIs, databases, or documents. Embedding language models convert this data into vectors and store it in a vector database.
- **Retrieve Relevant Information:** The user query is vectorized and matched with stored vectors. Example: A chatbot answering HR questions might retrieve leave policy documents and an employee's past leave record.
- **Augment the LLM Prompt:** The prompt is enriched with relevant retrieved data using prompt engineering techniques.
- **Update External Data:** To avoid staleness, documents and their embeddings should be periodically updated (real-time or batch processes).

RAG Pipeline

1. Prompt
2. Query database
3. Extract most relevant information
4. Combine prompt with retrieved information
5. Generate response

3 RAG with Knowledge Graphs

Integrating knowledge graphs into the RAG pipeline has demonstrated improved multi-hop reasoning.

- Train a LLM to extract a knowledge graph from unstructured text
- Insert this graph into the RAG pipeline as a structured intermediary

4 KGGen: Text to Knowledge Graph

KGGen is a system that uses language models to generate knowledge graphs from unstructured text.

Process Overview

1. **Entity and Relation Extraction**
   - Input text is processed to detect entities
   - A separate model extracts `(Subject, Predicate, Object)` triples
2. **Aggregation**
   - Unique triples are collected to form the graph
   - Normalization: all entities and edges are lowercased to avoid redundancy
3. **Clustering**
   - Cluster similar entities (e.g., USA, America, United States)
   - Deterministic unsupervised clustering algorithm (e.g., KNN, HCA)

Paper's Novel Idea: LLM as Judge

Handling Nodes

(1) The LLM receives clusters and performs binary classification to validate the groupings.

(2) If the input cluster passes the check, it is accepted as a valid group and labeled.

(3) The label is chosen to best represent the entire group.

(4) Steps 1–3 are repeated either for a fixed number of iterations or until no new valid clusters are found.

(5) The remaining unclustered entities are processed in batches. For each batch, the LLM checks whether any entities should be added to existing groups.

(6) This batch-to-cluster association step is repeated until all remaining entities are handled.

Handling Edges

- The same LLM-based classification is used to group edges.
- Prompts are modified to focus on relationships instead of entity clustering.