

Exploring 6 Knowledge Graph Methods with LlamaIndex and NebulaGraph

用 LlamaIndex 和 NebulaGraph 探索 6 种知识图谱方法

```
kg_index = KnowledgeGraphIndex.from_documents(
    documents=documents,
    storage_context=storage_context,
    max_triplets_per_chunk=15,
    service_context=service_context,
    space_name=space_name,
    edge_types=edge_types,
    rel_prop_names=rel_prop_names,
    tags=tags,
    include_embeddings=True,
)
```

0. VectorIndex向量检索

```
from llama_index import VectorStoreIndex

vector_index = VectorStoreIndex.from_documents(documents)
vector_query_engine = vector_index.as_query_engine()

response = vector_query_engine.query(query)
print(response)
```

1. KG vector-based entity retrieval

KG 基于向量的检索

```
# KG vector-based entity retrieval
kg_query_engine = kg_index.as_query_engine()
```

2. KG 基于关键词实体的检索

```
# KG keyword-based entity retrieval
kg_keyword_query_engine = kg_index.as_query_engine(
    # setting to false uses the raw triplets instead of adding the text from the corresponding nodes
    include_text=False,
    retriever_mode="keyword",
    response_mode="tree_summarize",
)
```

include_text=False: 查询引擎只用原生三元组进行查询, 查询不包含对应节点的文本信息;

只关注关系,可简化关系探索和潜在性能优化。

retriever_mode="keyword" 指采用关键词形式检索

`response_mode="tree_summarize"`: 返回结果是知识图谱的树结构的总结。以递归方式构建，查询作为根节点，最相关的答案作为叶节点。

```
# KG hybrid entity retrieval
kg_hybrid_query_engine = kg_index.as_query_engine(
    include_text=True,
    response_mode="tree_summarize",
    embedding_mode="hybrid",
    similarity_top_k=3,
    explore_global_knowledge=True,
)
```

3. KG 混合实体检索

KG hybrid entity retrieval

include_text=True: 查询引擎将在响应中使用相应节点的文本。

从关键字匹配的节点、深度搜索的节点和 top-k 相似性节点中检索文本，这可获取包含文本内容的全面信息。

`explore_global_knowledge=True`: 表示查询引擎应考虑知识图谱的全局上下文。

直接相关信息：苹果，苹果手机等

更广泛的信息：水果，水果的营养价值，水果种类等

4. 自定义组合查询引擎 (KG 检索和向量检索的组合)

Custom combo query engine

```
# Custom combo query engine
custom_query_engine = RetrieverQueryEngine(
    retriever=custom_retriever,
    response_synthesizer=response_synthesizer,
)
```

5. KnowledgeGraphQueryEngine [↗](#)

```
kgqe_query_engine = KnowledgeGraphQueryEngine(
    storage_context=storage_context,
    service_context=service_context,
    llm=llm,
    verbose=True,
)
```

[illegible]

KnowledgeGraphQueryEngine 返回了语法错误。根本原因似乎是生成的 Cypher 不正确，看来 KnowledgeGraphQueryEngine 的 Text2Cypher 能力还有待提高。

6. KnowledgeGraphRAGRetriever [↗](#)

```
# using KnowledgeGraphRAGRetriever
from llama_index.query_engine import RetrieverQueryEngine
from llama_index.retrievers import KnowledgeGraphRAGRetriever

graph_rag_retriever = KnowledgeGraphRAGRetriever(
    storage_context=storage_context,
    service_context=service_context,
    llm=llm,
    verbose=True,
)

kg_rag_query_engine = RetrieverQueryEngine.from_args(
    graph_rag_retriever, service_context=service_context
)
```