1. 知识图谱数据管理概述

知识图谱的目标是构建一个能够刻画现实世界的知识库，为自动问答、信息检索等应用提供支撑。因此，对知识的持久化存储并提供对目标知识的高效检索和更新是合格的知识图谱系统必须具备的基本功能，也是知识图谱数据管理的主要研究内容。

语言与图谱：知识载体

知识图谱是用于表示现实世界中存在的实体、事件或概念及其相互关系的有向图结构。实体可以是人、地点、组织等，边表示实体间的关系，如朋友、合作、所属等。实体和关系都可以有属性，如人物的姓名、年龄，关系的起始时间等。

知识图谱的结构化信息使其成为自动问答、推荐系统等应用的基础。语言是知识的载体，图谱则是对语言中所包含知识的结构化表达。

---

2. 符号化知识图谱数据管理

知识图谱数据模型

知识图谱的数据模型决定了数据管理所采取的方法和策略，对于存储管理、查询处理、查询语言设计均至关重要。常用的数据模型包括：

RDF图模型

- RDF三元组：$(\boldsymbol{s}, \boldsymbol{p}, \boldsymbol{o}) \in (\boldsymbol{U} \cup \boldsymbol{B}) \times \boldsymbol{U} \times (\boldsymbol{U} \cup \boldsymbol{B} \cup \boldsymbol{L})$
  - 例子：(`屠呦呦`, `出生日期`, `1930-12-30`)
- **RDF图**：由一组RDF三元组组成，表示实体及其属性和关系。

RDF（资源描述框架）是万维网联盟（W3C）制定的标准，用于表示信息的元数据模型。RDF使用三元组（subject-predicate-object）结构来描述资源及其属性和关系。RDF三元组的一个例子可以是（屠呦呦，出生日期，1930-12-30），这里屠呦呦是主体，出生日期是谓词，1930-12-30是客体。

属性图模型

- 属性图：是一个五元组：$\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{\rho}, \boldsymbol{\lambda}, \boldsymbol{\sigma})$
  - **V**：顶点的有限集合
  - **E**：边的有限集合
  - **ρ**: $\boldsymbol{E} \rightarrow (\boldsymbol{V} \times \boldsymbol{V})$：将边关联到顶点对
  - **λ**: $(\boldsymbol{V} \cup \boldsymbol{E}) \rightarrow \boldsymbol{Lab}$：为顶点或边赋予标签
  - **σ**: $(\boldsymbol{V} \cup \boldsymbol{E}) \times \boldsymbol{Prop} \rightarrow \boldsymbol{Val}$：为顶点或边关联属性

属性图模型广泛用于表示实体及其相互关系。顶点表示实体，边表示实体间的关系。每个顶点和边都可以有多个属性，例如顶点可以表示人物，具有姓名、出生日期等属性，边可以表示“朋友”关系，具有起始时间、亲密度等属性。

知识图谱数据的存储

知识图谱的数据存储可以基于表结构或图结构：

基于表结构的存储

- **三元组表**：简单直接的方式，但单表规模太大，查询和更新开销大。
- **类型表**：为每种类型构建一张表，减少数据冗余和空值，但需要多表连接，管理复杂。
- **层级类型表**：考虑类别层级结构，减少冗余和空值，提高查询效率，但仍需多表连接。

基于表结构的存储是传统的关系数据库方式，适用于结构化数据。三元组表方式虽然简单直观，但容易导致单表过大。类型表方式通过为不同类型的实体创建独立的表来解决这一问题，但增加了管理复杂性。层级类型表进一步考虑了实体类型的层次结构，减少了冗余数据，但查询时需要多表连接。

基于图结构的存储

- 将实体看作节点，关系看作带有标签的边，自然满足图模型结构。常见图数据库系统包括Neo4j、OrientDB、HyperGraphDB等。

基于图结构的存储方式更加自然地表示实体及其关系。节点表示实体，边表示实体间的关系。图数据库如Neo4j、OrientDB和HyperGraphDB等提供了高效的存储和查询功能，适用于复杂关系的表示和分析。

知识图谱数据的检索（查询）

- 声明式查询语言
    - **SPARQL**：用于RDF图的查询语言，支持数据操作（插入、删除、更新）和数据查询（SELECT、ASK、DESCRIBE、CONSTRUCT）。

SPARQL（SPARQL Protocol and RDF Query Language）是W3C为RDF数据开发的一种查询语言。SPARQL允许用户执行数据操作（插入、删除、更新）和数据查询（SELECT、ASK、DESCRIBE、CONSTRUCT）。SPARQL查询语句类似于SQL，但适用于RDF数据模型。例如，SELECT语句用于从知识图谱中获取满足条件的数据，ASK语句用于测试是否存在满足条件的数据。

- 过程式查询语言
    - **Cypher**：用于属性图的查询语言，常用于Neo4j。
    - **Gremlin**：Apache TinkerPop图计算框架的查询语言，支持图遍历和导航式游走。
    - **PGQL**：Oracle提出的图查询语言，支持正则路径查询。
    - **G-CORE**：LDBC图查询语言工作组设计的语言，支持图查询语言的可组合性。

Cypher是Neo4j使用的查询语言，类似于SQL，但专为图数据库设计。它支持节点和边的创建、查询、更新和删除操作。Gremlin是Apache TinkerPop框架的查询语言，采用遍历模式，适用于复杂图计算。PGQL是Oracle提出的图查询语言，支持复杂路径查询。G-CORE是LDBC图查询语言工作组设计的语言，旨在结合多种图查询语言的优点，提供更强的表达能力。

---

3. 参数化知识图谱（大模型）数据管理

大模型知识存储

大模型（如预训练语言模型）中，知识是以参数化的形式存储的。Transformer架构中的前馈神经网络（FFN）被认为是存储具体知识的Key-Value存储器：

- **Key层**：多层感知机MLP宽隐层，用于识别输入中的某种语言或知识模式。
- **Value层**：多层感知机MLP窄隐层，存储与模式相关的知识向量。

预训练语言模型（如BERT、GPT-3）中的知识存储在模型参数中。Transformer架构的前馈神经网络（FFN）模块被认为是存储知识的关键部分。FFN的Key层用于检测输入中的知识模式，Value层存储相应的知识向量。当输入匹配某种模式时，FFN输出对应的知识。

大模型的知识编辑（检索与更新）

知识编辑主要针对事实性知识的更新，方法包括：

- **全量微调+正则化项**：传统方法，更新所有模型参数。
- **超网络方法**：使用额外的神经网络生成参数更新量。
- **定向更新方法**：定位知识所在位置，直接修改相关参数。

最新研究进展

- **MEMIT**：同时更新大规模知识的方法，通过正规方程求解权重矩阵更新量，实现大规模知识编辑。
- **ROME**：更新单条知识的Key-Value目标，通过拉格朗日乘数法求解。

MEMIT（Mass-Editing Memory in a Transformer）是最新的大规模知识编辑方法，能够同时更新数万条知识。该方法通过正规方程求解权重矩阵的更新量，实现知识的高效更新。ROME（Rank-One Model Editing）是MEMIT的前作，主要更新单条知识，适用于较小规模的知识编辑任务。

自由形式和非内化的知识编辑

- **自由形式的知识编辑**：编辑常识知识和根据描述文本编辑实体知识。
- **非内化的知识编辑**：不改变模型参数，通过上下文学习或额外存储进行知识编辑。

自由形式的知识编辑任务涉及编辑常识知识和根据描述文本编辑实体知识，具有更高的灵活性和挑战性。非内化的知识编辑则不改变模型参数，而是通过上下文学习或使用额外存储进行知识更新，避免对模型结构的直接修改。

知识图谱与大模型协同更新（CogGPT）

利用知识图谱和非结构化文本增强大模型，实现问题理解和答案生成过程的可解释性，通过符号化编辑和检索增强实现知识图谱和大模型的协同进化。

CogGPT是一个结合知识图谱和大模型的框架，旨在增强大模型的解释性和准确性。通过将自然语言问题分解为图谱查询操作，CogGPT能够显式呈现大模型的问题理解过程，并利用知识图谱中的信息生成答案，缓解大模型的幻觉问题。

---

4. 代表性工具简介

符号化知识图谱数据管理工具

- **Neo4j**：基于Java的高性能图数据库，支持属性图模型和Cypher查询语言。
- **OrientDB**：文档-图混合数据库，支持多种模式和高级特性。
- **HyperGraphDB**：支持超图表示能力的存储系统。
- **InfiniteGraph**：分布式图数据库系统，基于Java开发。
- **InfoGrid**：开源互联网图数据库，提供额外组件构建基于图结构的网络应用。

Neo4j 是一种高性能的图数据库，支持ACID事务、Cypher查询语言，并提供丰富的开发工具和API。Neo4j的主要特点包括高效的图存储和检索、灵活的查询语言以及良好的可扩展性。它广泛应用于社交网络分析、推荐系统和知识管理等领域。

OrientDB 是一个文档-图混合数据库，支持图数据库的强大表示能力和文档数据库的灵活性。OrientDB支持多种存储模式（全模式、无模式、混合模式），并提供事务、快速索引、SQL查询等高级特性。

HyperGraphDB 是一个基于BerkeleyDB的开源图数据库，支持超图表示能力。超图允许边指向多个节点，提供比传统图更强大的表示能力。HyperGraphDB适用于复杂网络的表示和分析。

InfiniteGraph 是一个分布式图数据库，基于Java开发，支持面向对象的编程模型。InfiniteGraph需要作为服务项目进行安装，适用于大规模分布式环境。

InfoGrid 是一个开源互联网图数据库，提供丰富的组件，方便构建基于图结构的网络应用。InfoGrid核心是基于Java的图数据库项目，支持图存储和检索功能。

参数化知识图谱数据管理工具

- **EasyEdit**：适配多种大模型的知识编辑工具，支持T5, GPT-J, GPT-NEO, LLaMA等模型，以及超网络和定位编辑方法。

EasyEdit 是一个适配多种大模型的知识编辑工具，支持包括T5、GPT-J、GPT-NEO、LLaMA等在内的多个模型。EasyEdit提供了多种知识编辑方法，包括超网络方法和定位编辑方法，方便用户对大模型进行定向知识更新和优化。

---

5. 未来研究方向

持续知识编辑（解决灾难性遗忘问题）

研究如何在多次知识更新中避免模型的灾难性遗忘问题，保持已学习知识的稳定性。

持续知识编辑需要解决灾难性遗忘问题，即在多次知识更新中避免模型遗忘已学到的知识。通过设计更好的知识更新算法和优化策略，保持模型知识的连续性和稳定性。

双向知识编辑（逆转诅咒问题）

探索在知识编辑中双向更新的方法，即不仅能更新错误知识，也能撤销错误更新，恢复原始知识。

双向知识编辑涉及在知识更新中引入逆向操作，使得不仅能够更新错误知识，还能够撤销错误更新，恢复模型原始知识。这一研究方向旨在提高知识编辑的灵活性和可靠性。

鲁棒知识编辑（应对邻域扰动问题）

提高知识编辑的鲁棒性，确保编辑过程中不受邻域扰动影响，实现精准更新。

鲁棒知识编辑需要在面对邻域扰动时保持稳定性，确保知识编辑的准确性和一致性。通过优化算法和增加约束条件，提高知识编辑的鲁棒性。

应用扩展

- **多模态知识编辑**：扩展知识编辑至多模态数据，包括文本、图像、视频等。
- **智能体知识编辑**：研究在智能体中的知识编辑，实现智能体知识库的动态更新。
- **模型能力编辑**：不仅限于知识编辑，还包括模型能力的更新和增强。

多模态知识编辑 旨在将知识编辑扩展到多种数据类型，如文本、图像和视频，增强模型在多模态数据上的知识表示和推理能力。

智能体知识编辑 研究如何在智能体（如虚拟助手、机器人）中进行知识编辑，实现智能体知识库的动态更新，提升智能体的交互能力和响应准确性。

模型能力编辑 不仅关注知识的更新，还包括模型能力的提升和优化，如增加模型的推理能力、提高生成质量等，增强模型的整体性能。

总结

知识图谱数据管理是构建和维护知识图谱系统的关键，涵盖了数据模型、存储、检索、工具使用等多个方面。符号化和参数化两种知识图谱的数据管理方法和工具各具特色，研究和应用不断推进，未来将面临更多挑战和机遇。通过深入理解和持续探索，知识图谱的数据管理将为更多智能应用提供坚实的基础。