如何评估用于知识图谱处理的 NoSQL 数据库范式

Rosario Napoli
rnapoli@unime.it 墨
西拿大学 意大利

Antonio Celesti
acelesti@unime.it 墨
西拿大学 意大利

Massimo Villari
mvillari@unime.it
墨西拿大学 意大利

Maria Fazio
mfazio@unime.it
墨西拿大学 意大利

arXiv:2602.07612v1 [cs.DB] 7 Feb 2026

摘要

知识图谱（KG）处理在选择最优 NoSQL 数据库范式时面临关键的基础设施挑战
因为传统的性能评估依赖静态基准测试，无法捕捉真实世界知识图谱工作负载的复杂性。尽管大数据领域提供了大量比较研究，但在知识图谱场景中，数据库管理系统选择仍主要依赖临时决策，使从业者缺乏系统性指导，难以根据具体的知识图谱特征和查询需求匹配存储技术。本文提出了一个专用于知识图谱的基准测试框架，采用连通密度和规模，并引入一种以图为中心的指标——语义丰富度（SR），在四层查询方法论中揭示面向文档、图和多模型数据库管理系统之间的性能交叉点。我们在三个规模下对 FAERS 不良事件知识图谱开展实证评估，比较从简单筛选到深度遍历的各类范式，并提供以指标为依据、以证据为基础的指南，帮助根据图规模、连通性和语义丰富度选择合适的 NoSQL 范式。

CCS 概念： •**信息系统** → 数据分析；**决策支持系统**。

关键词：图数据科学、知识图谱、知识工程、图分析、性能分析。

1 引言

异构数据生成量呈指数级增长，推动了知识图谱（KG）作为组织和表示语义丰富信息的基础数据结构得到广泛应用 [29]。知识图谱通过纳入带类型的实体和关系以及丰富的元数据，扩展了传统图结构，从而能够以高层次的方式表示现实世界的动态变化。知识图谱广泛应用于生物医学研究、金融服务等不同领域，其日益重要的应用范围从企业数据湖 [8] 到语义推理系统，以及作为大语言模型（LLM）的真实数据来源，帮助缓解幻觉问题 [19]。

基于知识图谱的应用得到广泛采用，催生了图数据科学（GDS）这一学科，其重点是通过先进的图算法和基于知识图谱的机器学习（ML）技术，从图结构数据中提取洞见 [26]。然而，GDS 解决方案的高效部署取决于是否为数据存储和查询选择了最合适的数据库管理系统（DBMS）。事实上，与传统关系型数据不同，知识图谱具有独特的挑战，包括异构节点类型、不同的关系密度以及复杂的遍历模式，因此需要存储系统针对知识图谱的特定数据特征进行优化 [31]。本文旨在解决这一问题，为如何选择最符合特定知识图谱需求的数据库管理系统提供实用指南。

NoSQL数据库管理系统的生态提供了多种范式，例如面向文档、原生图和多模型方法，每种范式都针对不同的数据特征和查询模式进行了优化 [10]。具体而言，MongoDB 等面向文档的数据库[1], 专门用于基于属性的筛选和嵌套数据结构，而 Neo4j 等图数据库[2], 则专为关系遍历和模式匹配而构建。ArangoDB 3 等多模型系统试图连接这些范式，以灵活性为代价提供更大的适应性，并可能带来性能方面的权衡 [16]。然而，要确定哪种数据库管理系统范式在特定知识图谱特征和工作负载模式下表现更好，仍然是一项挑战；不断演进的大数据管理解决方案生态使这一问题更加复杂。低效的存储决策可能导致查询响应时间随着知识图谱规模的增长呈指数级恶化，可扩展性瓶颈可能限制分析能力，而开发开销则可能占用原本用于核心图数据科学任务的资源 [17]。

BDCAT '25，法国南特2025. ACM ISBN
979-8-4007-2286-8/2025/12
https://doi.org/10.1145/3773276.3774296

[1]https://www.mongodb.com/[2] https://neo4j.com/[3] https://arangodb.com/

此外，知识图谱具有异构性，从规模较小但语义丰富的数据集，到连接稀疏的大规模网络不等，这要求深入理解不同数据库管理系统范式在这一范围内的表现。

目前，知识图谱数据库管理系统的选择主要依赖临时决策，依据往往是开发者的直觉或供应商的营销宣传[14]。文献中关于这一主题的研究，通常基于合成数据集或单领域应用，对数据库管理系统进行性能分析。然而，这些方法并未专门针对知识图谱的需求，也未能为从业者提供系统性指导，帮助他们根据数据集的具体特征和查询需求选择合适的存储技术[12]。为弥补这一空白，我们分析了 NoSQL数据库管理系统模式在处理知识图谱工作负载时的性能。研究结果表明，在知识图谱领域，不同数据库管理系统范式之间的性能存在显著差异；这些发现可为图数据科学家提供具体且基于实证的基础设施决策指导，从而有望缩短开发时间，并提升各类知识图谱应用的系统性能。具体而言，本文的贡献如下：

1. 系统性的多数据集评估：我们使用具有不同特征和属性的真实世界知识图谱，对三种适用于知识图谱的关键 NoSQL 范式进行基于知识图谱、可复现的比较分析：面向文档的范式、原生图范式和多模型范式。

2. 渐进式查询复杂度分析：我们设计并实现了一个四层查询基准测试，逐步测试系统能力，从特定属性筛选到复杂的多跳遍历，从而能够识别性能交叉点，即最优数据库管理系统模式选择发生变化的节点。

3. 实用的选择指南：基于获得的实证结果，我们结合三项关键知识图谱数据集指标，提出以证据为依据的建议：

- **规模**：知识图谱的体量复杂度（$|V|+|E|$），会影响存储和索引开销。
- **连通性**：平均关系密度（$|E|/|V|$），会影响遍历和连接性能。
- **语义丰富度**：模式异质性，它结合了不同实体类型和关系类型的数量及其分布熵。

本文其余部分安排如下。第2节介绍全文使用的核心定义。第3节介绍用于知识图谱管理的 NoSQL 数据库管理系统背景，第4节讨论知识图谱背景下 NoSQL 数据库基准测试的相关工作。第5节对候选数据库管理系统模式进行比较分析。第6节介绍针对知识图谱提出的复杂度指标，第7节阐述方法论。数据集、基准测试设计和执行环境将在第8节中报告，随后在第9节呈现结果并进行讨论。最后，第10节总结全文并概述未来研究方向。

2 定义

本节介绍全文中使用的核心概念的形式化定义。

定义 2.1 (知识图谱)。 知识图谱定义为一个元组 $KG = (V, R, E, \ell_V)$，其中：

- V 是实体（节点）的集合；
- R 是关系类型的集合；
- $E \subseteq V \times R \times V$ 是带标签的有向边（三元组）的集合；
- $\ell_V : V \to C$ 是一个标注函数，它从集合 C 中为每个节点恰好分配一个类别标签。

定义 2.2 （知识图谱原子单元）。 知识图谱的原子单元是一个三元组[24]，由以下部分组成：

- 带标签的头节点 h；
- 带标签的尾节点 t；
- 一条带标签的有向边 r，从 h 指向 t。



图 1. 知识图谱原子单元。

定义 2.3 （冷启动条件）。 在没有缓存或先前优化的情况下执行查询：每个查询都从空内存开始，从而获得无偏且可复现的性能测量结果。

定义 2.4（热启动条件）。 预热后执行查询：使用先前查询产生的缓存和优化，反映稳态性能，通常响应速度更快。

3 知识图谱管理的 NoSQL 数据库管理系统背景

知识图谱异构的结构和语义要求采用高度灵活的方法进行存储和信息检索。这些技术不仅必须考虑多种数据类型共存的情况，还必须具备可扩展的基础设施，以高效处理动态的图增长和波动的查询负载[12]。为满足这些需求，人们开始转向受 NoSQL 数据库启发的范式；这些数据库专为克服传统关系型数据库系统在性能和可扩展性方面的限制而设计。

在这种情况下，在知识图谱领域选择 NoSQL 数据库管理系统，并非单纯由性能驱动的决策，而是

还与语义表达能力有关，因为每种范式对多跳遍历和异构模式表示的支持程度各不相同。多项研究分析了不同数据库管理系统架构如何对数据进行建模和管理，尤其关注涉及从多个来源进行数据集成的场景 [11]。在 NoSQL 解决方案中，可区分出四个主要类别：

- **键值存储**，它们将信息作为简单的键值对进行管理；
- **面向列的存储**，它们按列存储数据，从而能够选择性地仅访问与查询相关的属性；
- **面向文档的存储**，它们管理半结构化实体，例如 JSON；
- **面向图的存储**，它们以节点和边的形式对数据进行原生建模，以显式表示关系[27]。

每一类在模式灵活性、可扩展性以及对复杂查询的支持之间提供了不同的平衡。因此，底层系统的选择在知识图谱管理中发挥着重要作用，因为它直接影响图数据科学（GDS）工作流中知识抽取与推理的效率、可扩展性和质量 [13]。此外，结合多种NoSQL范式的混合方法不断兴起，进一步增加了选择过程的复杂性 [21]。

为了更好地把握这些差异，可以沿与其内部模式模型相关的两个关键轴对NoSQL 数据库管理系统进行映射：可扩展性与复杂性（见图 2）。这种映射有助于理解每种范式如何契合知识图谱建模需求，即大规模数据与语义丰富、相互关联的结构往往需要共存。

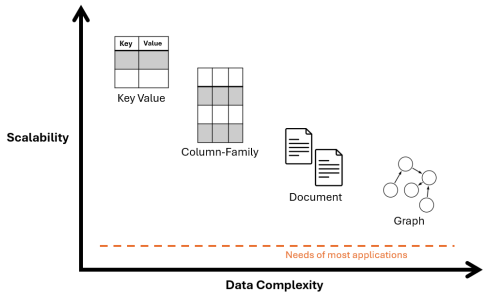


图 2. 按可扩展性与复杂性划分的NoSQL 范式分类。

基于这一分类，我们现在进一步考察主要的NoSQL范式，首先从键值存储谈起。键值存储在高性能键查找方面尤为高效，尤其适用于分布式环境。然而，它们在知识图谱场景中的应用有限，因为它们不原生支持关系或语义上下文。在这类系统中表示知识图谱通常需要在键或值中手动编码图语义，这种方法很快就会变得低效。此外，如果不在外部重建图，执行多跳查询或模式匹配就不切实际，从而降低了它们在知识图谱相关任务中的可用性 [4]。

列族数据库（如Cassandra[4], ）能够存储带有可变列的宽行，这些宽行可用于编码邻接矩阵以表示图。尽管具有这一优势，它们仍然缺乏用于高效边遍历和语义推理的原生机制 [7]。因此，在表达性关系查询方面，列族模型与键值存储存在类似的局限性。

相比之下，图数据库和文档存储因能够处理异构信息并针对特定查询模式进行优化，而成为尤其适合知识图谱管理的选择 [6]。尤其是，面向文档的数据库支持嵌套实体和跨文档引用，从而能够在单个文档或集合中表示局部图片段。图数据库

，另一方面，图数据库专门针对高效存储和查询相互连接的元素网络而设计，但并不适合处理海量数据。

最后，多模型数据管理解决方案的出现又增添了一层复杂性，因为它们旨在通过单一框架整合多种NoSQL 范式的优势，进一步增加了知识图谱管理决策过程的复杂程度。

总而言之，NoSQL数据库是知识图谱表示的重要基础，因为它们能够处理异构且不断演变的数据集，并具备在分布式环境中的可扩展性，从而支持对复杂结构进行高效推理。尽管如此，这些优势能否发挥出来，很大程度上取决于是否选择了最合适的范式，因为每种范式都有各自的权衡，会影响知识图谱性能、可扩展性和整体可用性。

4 相关工作

NoSQL 数据库系统的性能评估已成为一个重要的研究领域，其驱动力在于了解不同存储范式如何处理多种工作负载和数据特征。尽管针对传统 NoSQL 系统领域已有大量基准测试研究，但 NoSQL 领域中的知识图谱应用仍必须分析与异构架构和专用优化策略相关的具体权衡 [2]。

然而，现有的大多数性能研究仅关注通用数据库工作负载，而非知识图谱管理的特定需求 [1]。具体而言，知识图谱带来了不同的挑战，包括异构节点类型、变化的关系密度、复杂的遍历模式，

[4]https://cassandra.apache.org/index.html

以及需要专门评估方法的语义推理要求 [5]。

此外，知识图谱社区尚未完全从经典方法转向更现代的 NoSQL 方法，例如资源描述框架（RDF）——这是一项 W3C 标准，以主谓宾三元组表示知识—— [30]。这种不完全的转变造成了碎片化的局面，从业者缺乏明确的指导，难以选择最合适的数据模型和数据库技术。因此，图数据科学家通常依据自身的既有知识或专业经验做出架构选择，导致不同项目之间的实践不一致。

这一差距给知识图谱的可扩展部署带来了重大挑战，因为次优的数据库管理系统架构决策可能导致性能问题，而这些问题会随着数据集规模的增长和查询复杂度的提高而加剧。这迫使从业者投入大量精力评估和优化数据库系统，而非专注于图算法开发和分析洞见，造成资源分配低效，并限制了图数据科学工作流程中的创新。

4.1 NoSQL 性能基准测试

传统的 NoSQL 基准测试方法主要针对通用数据模型，着重考察增删改查操作、吞吐量指标以及统一工作负载下的可扩展性 [24]。早期的比较研究使用与领域无关的大规模数据集，为文档存储、键值系统和图数据库提供了性能基线。然而，这些方法并非为知识图谱场景设计，因为它们忽略了此类场景的语义丰富度和关系复杂性 [25]。

近期的基准测试工作也开始纳入更贴近现实的工作负载，包括多模型场景和异构查询模式。尤其是，在执行复杂关系型查询时，不同 NoSQL 范式之间的性能差异显著；对不同范式的比较分析也揭示了它们在遍历效率和模式匹配能力方面存在实质性差异 [6]。然而，这些研究仍受限于对通用大数据数据集的依赖。因此，它们未能捕捉真实世界知识图谱场景所具有的异构性、语义丰富度和关系复杂性，限制了其研究结果在面向知识图谱的用例中的适用性。

4.2 知识图谱存储系统

知识图谱管理与存储系统性能的交叉领域是一个相对较新的研究方向。早期工作主要聚焦于 RDF 三元组存储及其针对 SPARQL 查询的优化，并为语义网应用建立了性能模式 [12]。特别是，传统的知识图谱表示方法主要侧重于 RDF 三元组存储和基于关系模式的模型，强调 SPARQL 查询性能和推理能力，这些方面塑造了早期关于知识图谱存储与查询的大部分研究 [15]。尽管 NoSQL 系统提供了其他数据建模策略，但关于这些系统如何处理知识图谱典型的结构复杂性和语义复杂性，相关研究仍然有限。

尤其是，关于数据集的内在属性（如语义、连通密度和可扩展性要求）如何影响 NoSQL 范式在知识图谱表示中的适用性，仍存在一个关键的研究空白。现有研究往往只是复现传统的三元组存储设置，因而未能充分探讨数据特征与 NoSQL 存储策略之间的相互作用 [9]。然而，针对多个具有不同特征的真实数据集开展系统性比较研究仍然有限，导致从业者在选择存储系统时缺乏基于证据的指导。

5 用于知识图谱表示的数据库管理系统模式

为了更好地了解当前数据库技术如何支持知识图谱管理的特定需求，我们将聚焦于通常被认为在表示知识图谱方面更具表达力且更为成熟的数据模型。基于这些原因，我们不讨论键值数据库和列式数据库。尽管这些模型在一定程度上能够存储类图结构（例如，通过编码邻接关系或实体属性），但它们缺乏用于边遍历、语义推理和表达性关系查询的原生机制，因此本质上不太适合高级推理和高层次的数据互联。事实上，现有关于知识图谱数据管理的文献大多聚焦于明确设计用于支持异构知识结构和面向图的推理的解决方案 [32]。图数据库、面向文档的系统和多模型方法始终被视为最有前景的范式，因为它们提供的机制能够兼顾灵活性、语义丰富度和可扩展性。

这一选择使我们能够探究最适合的 NoSQL范式如何应对核心挑战：管理高度互连且异构的数据，其中节点可能属于不同的类并具有众多属性。这也有助于我们更清晰地了解它们在实际知识图谱场景中的各自优势与权衡。具体而言，我们考察了三种具有代表性的系统：图数据库 Neo4j、基于文档的 MongoDB，以及混合型图文档数据库 ArangoDB。它们分别代表了知识驱动场景中常用的不同 NoSQL范式。以下各小节将结合知识图谱特定需求，探讨它们的架构基础与设计选择。

5.1 Neo4j：面向知识图谱的原生图建模

Neo4j 被广泛认为是图数据管理领域最成熟的技术之一，并且常用于知识图谱研究和应用 [18]。它基于图模型，能够将实体及其语义关系直接表示为节点和边，每个节点和边都具有键值属性。这种原生图结构尤其适合捕捉知识图谱固有的表达能力与连通性。

Neo4j 的一项标志性特征是其声明式查询语言 Cypher，该语言专为处理复杂图模式而设计。它支持多跳遍历和子图匹配，这些都是知识图谱探索与推理中的基础操作 [20]。与通用 NoSQL 系统不同，Neo4j 专为处理以图为中心的工作负载而构建，因此非常适合需要对丰富本体进行建模、推断隐式知识以及遍历深层嵌套关系的领域。然而，这种架构也带来了可扩展性限制：图数据库管理系统在处理大规模数据集时，性能会显著下降；随着图规模和复杂度的增加，查询执行时间呈指数级增长（图 2） [17]。

5.2 MongoDB：面向半结构化知识的灵活文档模型

MongoDB 是使用最广泛的面向文档 NoSQL 数据库，基于半结构化的知识表示提供灵活的模式设计 [28]。尽管它并不原生支持图操作，但其采用的类 JSON 格式允许嵌套文档并构建层次化结构，因此可以调整这种结构，在数据库中捕获部分类似知识图谱的信息。

具体而言，它将数据组织成文档集合，每个文档都可以嵌入多层相关信息。这种方法适用于知识具有局部结构、且数据关系可编码在文档层次结构中的应用场景 [23]。尤其是，文档模型可视为其他数据模型的超集，其中包括键值、关系型、对象和图模型。具体而言，它在聚合管道中提供了 graphLookup 阶段，支持对树和层次化数据等类似图的结构进行递归遍历。该功能通过反复查找指定关系中的文档，直至满足终止条件，从而支持探索相连文档。然而，graphLookup 无法提供完整的图算法，也不具备原生图数据库通常拥有的优化遍历性能。

归根结底，这种以文档为中心的架构在图密集型操作方面存在固有局限；与原生图数据库相比，MongoDB 处理复杂图遍历的性能会显著下降，尤其是在处理高度互连的知识图谱拓扑结构时，这类拓扑需要大量跨文档关系和多跳查询（图 2）。

5.3 ArangoDB：通过多模型设计实现统一访问

ArangoDB 引入了多模型架构，在单一集成引擎中原生支持两种不同范式：文档和图。这种设计为需要不同类型数据访问的应用提供了统一接口，无需采用独立的数据库系统或多语言持久化。

在知识图谱的语境下，ArangoDB 原生支持图结构，同时允许采用面向文档的存储。其灵活性使其适用于混合场景：部分知识具有高度的关系性，而其他信息则更适合表示为属性或嵌套实体 [22]。其内部数据表示基于 JSON，数据被分组到集合中，以区分文档类型和边类型。

它使用自有的查询语言 AQL（Arango 查询语言），该语言受 SQL 启发，并支持跨多个模型执行操作。尽管 AQL 支持图遍历，但与 Cypher 相比，它通常更加冗长，语义聚焦程度也较低，因此在以图为主的场景中，用户体验不够直观。

我们的比较分析表明，所考察的数据库管理系统解决方案没有哪一种能够被视为绝对意义上的“完美”方案。这些各不相同的优势与劣势说明，最合适的数据库管理系统的选择严格取决于知识图谱的结构和语义特征，也凸显了在知识图谱管理中采用自适应、系统化方法来支持选择过程的必要性。

6 用于评估 NoSQL 数据库管理系统的知识图谱复杂度指标

我们对文献中相关工作的回顾表明，现有基准测试并未考虑关键的知识图谱特有因素。对NoSQL 数据库管理系统的传统评估主要强调总体吞吐量、延迟和可扩展性指标，这些指标在通用数据库场景中颇具价值，但不足以刻画知识图谱的复杂性。具体而言，语义关系可能差异很大，从而产生不平衡的分布：图中某些区域密集且语义丰富，而其他区域则稀疏且以属性为中心。这种异质性对数据库管理系统性能具有显著影响。例如，若将原生图存储中的查询应用于大规模属性负载或节点类型的长尾分布，其计算开销可能会变得很高，而文档存储则可能表现更优

在属性密集型场景中，但在深度遍历或关系密集型查询方面则较为吃力。这些考量凸显了采用系统化方法的必要性，即将可度量的知识图谱属性与所观测到的系统性能联系起来。

为弥补这一空白，我们旨在评估知识图谱的三个相互正交且可量化的维度：规模、连通性和语义丰富度，每个维度都代表知识图谱复杂度的一个特定方面。这些维度使我们能够设计一组实验，系统地探索不同的NoSQL范式如何应对知识图谱拓扑结构、模式分布和语义配置的变化。通过将性能结果映射到这一三维空间中，我们可以确定一种范式开始优于另一种范式的交叉点。这为从业者选择数据库管理系统提供了基于证据的指导。

给定一个如前所述的知识图谱 $KG = V,\ R,\ E,\ \lambda_V$，我们引入以下维度：

定义 6.1（规模）。给定 KG，规模维度 $S(KG)$ 衡量其体量复杂度：

$$S(KG) \ = \ |V| + |E|.$$

其中，$|V|$ 和 $|E|$ 分别表示实体数和边数。规模可用于衡量存储需求、索引开销和内存消耗。

定义 6.2（连通密度）。给定 KG，连通维度 $CD(KG)$ 用于量化平均关系密度：

$$CD(KG) \ = \ \frac{|E|}{|V|}.$$

较高的 $CD(KG)$ 表示图的互联程度更高，从而增加遍历复杂度和连接操作。

定义 6.3（语义丰富度）。给定 $KG = V, R, E, \lambda_V$，语义丰富度 $SR(KG)$ 通过结合两个方面，衡量图模式的多样性和表达能力：不同元素的数量及其分布的均匀性。

首先，令：

- $|C|$ = 实体类型数
- $|R|$ =关系类型数量。

我们通过以下方式衡量类型多样性：

$$D_{\text{types}} = \log |C| + \log |R|.$$

数值越大 D_{types}，表示节点和边的种类越多。

接下来，我们使用香农熵来衡量分布均匀性，香农熵量化了数据分布中不确定性或随机性的程度 [3]。形式化表示为：

$$p(c) = \frac{\left|\{v \in V : \lambda_V(v) = c\}\right|}{|V|}, \quad p(r) = \frac{\left|\{(u, r, v) \in E\}\right|}{|E|}.$$

其中，$(u,\ r,\ v)$ 是知识图谱的原子单元。于是：

$$H(C) = -\sum_{c \in C} p(c)\ \log p(c), \quad H(R) = -\sum_{r \in R} p(r)\ \log p(r).$$

熵越高，类型或关系的分布越均匀，从而增加查询复杂度。

最后，我们将这些组成部分结合起来：

$$SR(KG) = D_{\text{types}} + H(C) + H(R)$$

具体而言，当图具有以下特征时，$SR(KG)$ 会增大：

- 多种不同的节点和边类型（D类型繁多）；
- 和/或这些类型的分布均衡（熵高）。

这些维度共同实现了以下目标：

1. **数据集选择：** 选择覆盖（$S,\ CD,\ SR$）空间不同区域的评估图，确保涵盖真实世界的知识图谱场景。2. **性能解释：** 将观察到的数据库管理系统性能映射到特定的（$S,\ CD,\ SR$）组合，从而揭示一种范式优于另一种范式的交叉点。3. **泛化：** 将性能趋势与底层知识图谱特征相关联，推导出适用于不同数据集的选择指南。

通过为实验方法奠定基础，我们将临时性的数据库管理系统选择转变为系统且可复现的流程，将可量化的知识图谱属性与实证观察到的性能模式联系起来。

7 评估方法论 Y

我们提出了一套可复现的工作流程，用于评估不同规模下 NoSQL 数据库管理系统范式在知识图谱工作负载上的表现。该方法论在规模逐步增大的数据集上重复执行相同流程，从而能够系统地探索可扩展性-复杂度性能空间（图 2）。通过这种方法，我们可以观察三种数据库范式在不同条件下的表现，并识别数据量呈指数增长时的性能拐点。该工作流程包含五个依次执行的步骤：数据准备、系统导入、指标计算、查询执行和结果汇总，并在所有数据集上以相同方式执行。在下文各小节中，我们将详细分析评估方法，介绍该方法的各个组成部分。

7.1 数据准备

输入数据集以图格式提供，其中包含实体和关系结构。我们开发了自动化连接器，可转换原始图表示并将其转换为所需的目标格式。具体而言，我们从原生图结构生成适用于面向文档和多模型系统的 JSON表示。

7.2 数据集扩展策略

为了评估系统在逐渐增大的工作负载下的可扩展性，我们采用迭代复制方法。从大小为 X 的原始数据集开始，将整个数据集复制并与自身合并，使其在每一步的大小都翻倍。经过 n 次复制后，数据集大小遵循指数增长函数：

$$n \text{ 次复制后的数据集大小 } = X \times 2^n$$

7.3 系统导入

每个数据集在导入三种范式之前，都会经过特定格式转换。原生图系统直接摄取原始图格式，而面向文档和多模型系统则通过自动化连接器，采用两阶段摄取方法处理数据。转换过程会将原始图结构转换为两种独立的 JSON 表示形式：一种包含各个节点类，另一种包含用于编码主语-谓语-宾语结构的 RDF三元组，适用于面向文档和多模型范式。

7.4 指标计算

数据摄取后，我们立即在导入的数据上计算三个知识图谱复杂度指标：

- $S(KG) = |V| + |E|$;
- $CD(KG) = \frac{|E|}{|V|}$;
- $SR(KG) = D_{\text{类型}} + H(C) + H(R)$.

7.5 查询执行与结果聚合

我们执行一套包含四个查询层级的测试，以检验不同的系统能力：

- **第1层：** 复杂的单属性筛选。
- **第2层：** 复合多属性选择。
- **第3层：** 有限的（1–2跳）模式匹配。
- **第4层：** 深度（3+跳）多跳遍历。

每个查询分两个阶段运行：

1. **冷启动：** 每次运行前清除缓存。共运行 31 次。2. **热启动：** 在 31 次运行期间保留缓存。

每次运行时，我们测量**执行时间**（毫秒），最后将各个查询的结果汇总为图表，展示不同数据库管理系统在规模不断增大的数据集上执行相同查询时的性能，并绘制均值及 95% 置信区间。

8 实验结果

本节详细介绍了遵循评估方法的实验流程，涵盖测试的数据库系统、数据准备与扩展、数据摄取流程、计算指标、查询类型，以及每种查询的测量性能。所有实验均在一台配备 Intel Core Ultra 5 处理器、16 GB 随机存取存储器和 Intel Arc 系列 GPU、运行 Windows 11 Pro 的单台工作站上进行。

8.1 数据集准备

实验基于一个真实世界数据集开展，该数据集源自美国食品药品监督管理局（FDA）不良事件报告系统（FAERS）[5]。FAERS 是 FDA 建立的数据库，用于支持对获批药物进行上市后安全性监测。具体而言，该系统收集由医疗专业人员、消费者和制造商提交的不良事件及用药错误报告。该数据集被建模为 Neo4j 中的知识图谱[6]，包含以下类：病例、药物、反应、结局、报告来源、治疗和年龄组（图 3），涵盖八种主要实体，约有 11,000 个节点和 14,500 条边。

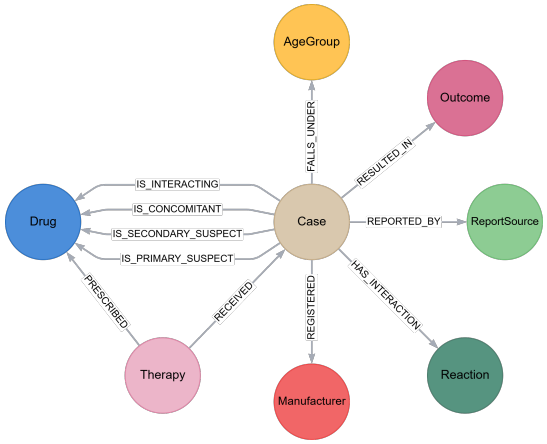


图 3.FAERS 知识图谱的架构。

数据准备首先将整个图导出为中间 CSV 文件，每个节点标签对应一个文件，每种关系类型也对应一个文件。每条 CSV 记录都包含元素唯一的 Neo4j 标识符，以及其属性的 JSON 序列化表示。序列化程序能够处理嵌套字典和列表。基于这些 CSV 文件，生成了两组 JSON 文档：

- 对于 MongoDB，每个节点文档都使用 Neo4j ID，并包含一个属性字段，其中存放反序列化后的 JSON 对象；关系文档包含：id、from、to 和 "properties";
- 对于 ArangoDB，全局查找表将 Neo4j ID 映射到对应的标签，从而能够按 <集合>/<_键>的格式构建 key、from 和 to 字段，并将所有原始信息保存在 "properties" 字段中。

[5]https://open.fda.gov/data/faers/ [6] https://github.com/neo4j-graph-examples/healthcare-analytics

8.2 数据集扩展策略

从原始 FAERS 出发，我们在每一步都将整个图复制一份并与自身合并，使其呈指数级增长。具体而言，我们选取了三个具有代表性的规模：

- **1×**：原始数据集，基准规模。
- **3 重复（8×）**：中等规模场景，同时对数据摄取和查询性能进行压力测试。
- **7 重复（128×）**：大规模场景，用于挑战系统极限并观察可扩展性趋势。

8.3 系统导入

数据摄取层负责协调将准备好的 JSON表示通过各自的原生驱动程序传输到目标数据存储。对于 MongoDB，Python 客户端会连接到指定数据库，并遍历每个包含节点和关系文档的集合。

在 ArangoDB 数据处理流程中，数据摄取脚本利用 ArangoDB Python 驱动程序连接到文档集合和边集合。节点文档会加载到各自的顶点集合中，边文档则根据_from和_to字段加载到指定的边集合中。

8.4 指标计算

数据摄取完成后，我们会针对每种规模配置计算知识图谱复杂度指标。表 1 汇总了在所有规模配置下计算得出的语义丰富度指标和连通性度量，详细展示了不同粒度级别下知识图谱的结构和语义特性。

表 1. 不同规模下 FAERS 知识图谱的语义丰富度指标和连通密度（CD）。

规模	节点	关系	数据类型 HC	HR	SR	CD
1×	14,000	11,000	4.48	1.39	2.04	7.91 0.79
8×	112,000	88,000	4.48	1.39	2.04	7.91	0.79
128×	1,792,000	1,408,000	4.48	1.39	2.04	7.91	0.79

8.5 查询执行与结果聚合

一个 Python 脚本驱动 MongoDB、ArangoDB 和 Neo4j 中四个查询层级在 31 次冷启动与温启动运行上的执行。聚合步骤读取查询执行阶段生成的 CSV 文件，每个文件包含冷启动与热启动执行的逐次运行计时和元数据。一个 Python 脚本加载这些 CSV，计算 31 次运行的平均执行时间及 95% 置信区间，将其合并并绘图。接下来，我们分析各个查询及其聚合结果。

8.5.1 第 1 层：复杂的基于属性的筛选（图 4 和图 5）。在第一层中，我们通过结合年龄、职业和性别条件来缩小病例报告的范围。系统会扫描每条病例记录，并选出符合以下条件的记录：患者年龄在 60 至 90 岁之间，报告来自医院工作人员，且性别标记为女性或男性。

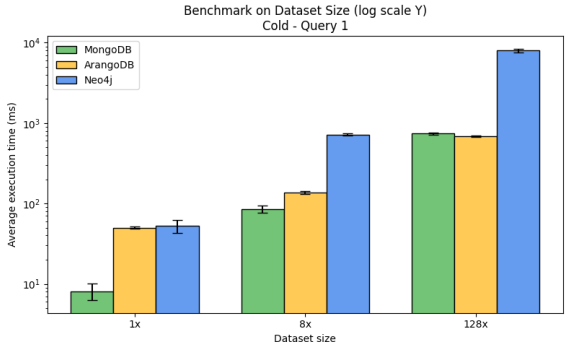


图 4. 各系统上查询 1 的冷启动性能。

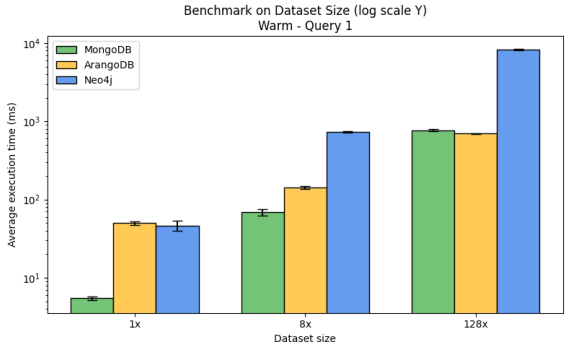


图 5. 各系统上查询 1 的温启动性能。

8.5.2 第2层：筛选与一跳连接（图 6 和图 7）。在此场景中，系统首先识别医院工作人员提交的所有病例报告。对于其中每个病例，系统沿所有将病例关联到其对应年龄组的关系进行遍历。仅保留标记为“儿童”或“成人”的年龄组。

8.5.3 第3层：病例到制造商、年龄组和药物的多重连接（图8和图9）。在这里，查询首先查找所有至少有一种处方药的病例。然后，系统会沿着另外两条连接继续查询每个病例：一条指向药物的制造商，另一条指向患者的年龄组。系统会为每个病例汇总药物、其生产商以及患者的年龄段。

8.5.4 第4层：深度上下文病例及所有相连实体（图10和图11）。最终查询探索每条病例记录周围的完整邻域。从每个病例节点开始，系统检索所有直接相连的实体：制造商、年龄组、结局、报告来源、药物、反应和治疗。

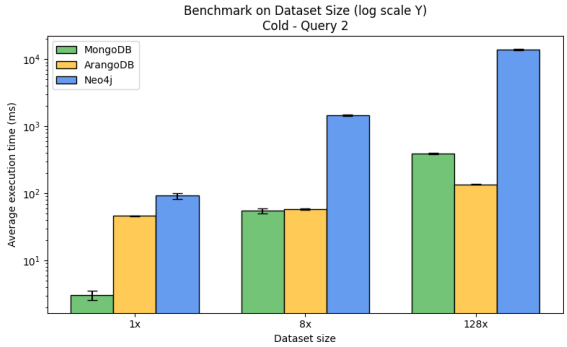


图 6. 各系统查询 2 的冷启动性能。

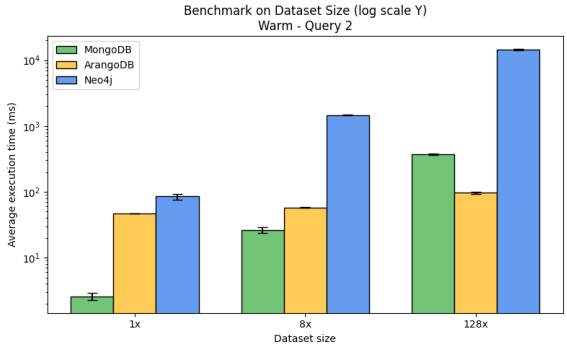


图 7. 各系统查询 2 的热启动性能。

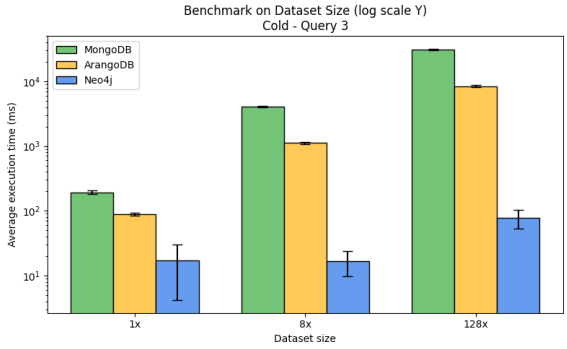


图 8. 各系统查询 3 的冷启动性能。

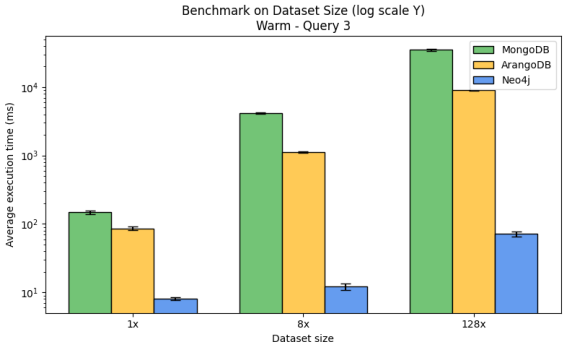


图 9. 各系统上查询 3 的热启动性能。

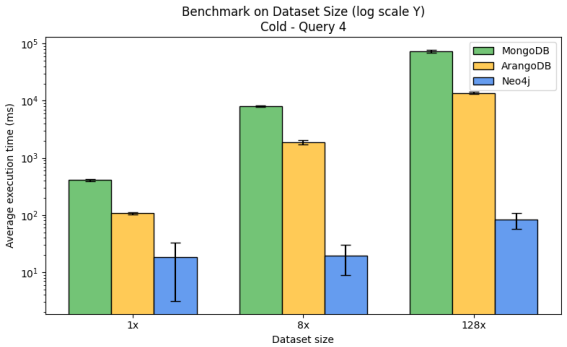


图 10. 各系统上查询 4 的冷启动性能。

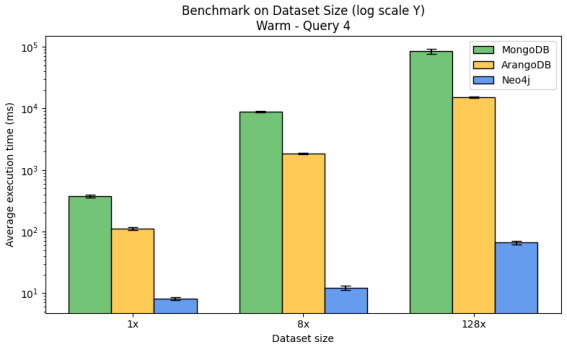


图 11. 各系统上查询 4 的热启动性能。

9 结果讨论

实验结果综合展示了 NoSQL范式在一系列查询中的表现，同时考虑了 FAERS 知识图谱的语义和结构属性。

对于 Tier1 属性筛选，MongoDB 的面向文档的引擎始终具有最低的响应时间（图 4、图 5）。在此设置下，SR$_{KG}$ = 7.91 和 C$_{KG}$ = 0.79意味着模式异构性适中，但连通性有限，因此能够高效执行单集合扫描和索引查找。随着数据集规模从 1× 增长到 128×，节点和边数量虽呈线性增长，但索引性能并未受到影响，这证实了文档存储是低跳数、高吞吐量查找的首选。此外，当某些子图包含丰富的实体但连接仍然稀疏时，

例如，具有大量元数据的异常药物或结果节点，面向文档的系统使数据科学家能够深入进行属性分析，而不会产生图遍历开销。然而，在最后一个规模步骤（128×），ArangoDB 开始显现出微弱优势，这表明在极端数据量下，其混合存储模型或许能更好地平衡索引分辨率与轻量级边导航。

进入引入跳连接的第2层后，我们确认了此前的趋势，并观察到随着数据集大小增加，性能优势转向多模型数据库（图6、图7）。在这里，恒定的 $D_{types} = 4.48$, $H(C) = 1.39$ 和 $H(R) = 2.04$ 产生了均衡的 SR_{KG}，这对纯文档引擎构成了挑战，但仍未达到深度遍历的阈值。多模型系统交替执行文档检索与原生图导航，能够承受中等程度的 C_{KG} (0.79)，而无需承担完整的图索引开销。这种混合能力对于呈现异构连接性的知识图谱尤其有利：其中既有稀疏但信息丰富的节点区域，也有连接密集的簇，从而能够高效地进行局部遍历，同时保持整个图的以属性为中心的性能。

在第3层和第4层的深度遍历中，随着数据集规模扩大，原生图平台展现出更优异的热运行性能（图8–图11）。高SR_{KG}和C_{KG}会加剧随机I/O需求并增加多跳路径枚举的复杂度，而Neo4j通过邻接优化存储和原生遍历规划来应对这些需求。冷启动测量（图10）反映了置信区间中的初始缓存预热情况，但热运行曲线表明，对于连接丰富且类型多样性高的图（例如需要全面遍历的图），原生图引擎能够以最高效率执行深度语义查询。

图5、图7和图11的热运行图中观察到的交叉点，界定了SR_{KG}–C_{KG}空间中的经验边界。通过在数据摄取时计算SR_{KG}、D_{types}, $H(C)$、 $H(R)$和C_{KG}，并将查询划分为四个层级，从业者便可将本研究的发现用作决策框架。在实践中，图数据科学家应考虑以下建议：

- **面向文档的引擎：** 适用于属性信息占主导且边较为稀疏的数据集。面向文档的引擎利用嵌套文档结构和高效的字段索引，因此非常适合单属性筛选和小规模查找。这类知识图谱的表示方式是：节点通过丰富且详细的属性承载大部分信息，而节点之间的关系较为稀疏，语义也较弱。因此，其结构连接较松散，但描述性很强。

- **多模型解决方案：** 它们非常适合 SRKG 处于 5–10 范围且 CKG $\approx$ 0.8 的图，反映出混合型连通性模式，其中稀疏但信息密集的区域与密集簇并存。其混合架构将文档检索与有针对性的图导航无缝结合，无需承担完整的原生图开销，即可高效完成一至两跳遍历和属性连接操作。这类知识图谱最适合描述为结构混合型：既包含语义密度高的区域，也包含由属性丰富但连接较弱的节点主导的区域。

- **原生图平台：** 适用于 $SR_{知识图谱} > 8$或 $C_{知识图谱} > 0.8$，这类知识图谱包含语义丰富、彼此互连的子图，需要进行深入的多跳遍历。原生图平台提供针对邻接关系优化的存储和原生遍历规划，从而最大限度地减少随机输入/输出，并利用内存路径探索来支持复杂推理和递归查询模式。这类知识图谱在语义上具有同质性且连接密集，大多数区域关系丰富、相互联系紧密。

10 结论

知识图谱管理需要的不只是原始吞吐量，还要求存储模式、图语义和连通模式之间经过审慎协调。我们提出的语义丰富度和连通密度指标揭示了不同 NoSQL 范式在处理知识图谱固有的多样性和稀疏性方面存在显著差异。通过在 FAERS不良事件图上对 MongoDB、Neo4j 和 ArangoDB 进行实证评估，我们发现了明确的权衡，为从业者提供了有价值的见解。

通过在指标驱动的分析与实证性能观察之间建立具体联系，本研究将传统上临时性的数据库管理系统选择过程转变为系统化、基于证据的方法论。为从业者提供直接针对其知识图谱特定语义特征和查询需求的模式选择具体指南，超越基于直觉的决策，转向数据驱动的基础设施选择。

展望未来，本研究为自动化通用框架奠定了基础，使其能够根据任意知识图谱数据集动态调整存储模式。未来的工作旨在通过对知识图谱特征进行严格分类，开发能够自动选择最优范式的智能系统。这种向智能化、自适应数据库选择的演进，是让不同领域的从业者更便捷、高效地使用知识图谱技术的自然下一步。

致谢

Rosario Napoli 是一名博士生，就读于国家人工智能博士项目第 XL 届健康与生命科学课程。本研究部分由“欧洲开源深度学习平台：专为嵌入式硬件与欧洲而设”项目资助，该项目获得了 Horizon

Europe“开源深度学习平台：专为嵌入式硬件与欧洲而设”项目（资助协议，项目编号 101112268 - NEUROKIT2E）、“网络空间安全与权利（SERICS）”合作项目（PE00000014），由欧盟“下一代欧盟”意大利大学与研究部国家复苏与韧性计划资助。具体而言，该研究获得了 SERICS 合作项目的支持，资助项目包括 FF4ALL（CUP D43C22003050001）和 SOP（CUP H73C22000890001）；还获得了意大利大学与研究部（MUR）“国家利益研究项目（PRIN-PNRR）”的支持，资助项目为“面向智能可持续环境的按需服务云连续体”（CUP：J53D23015080001-IC：P2022YNBHP）；以及意大利大学与研究部（MUR）“国家利益研究项目（PRIN）”计划的支持，资助项目为“远程康复即服务（TRaaS）”（CUP J53D23007060006）。

参考文献

[1] Sithursan等，2024年。SM3-文本转查询：合成多模型医疗文本转查询基准测试。（2024）。doi:10.48550/arXiv.2411.05521 预印本见arXiv。[2] Camelia-Florina Andor 和 Bazil Pârv。2018 年。NoSQL 数据库性能基准测试——案例研究。布贝什-博雅伊大学学报.- 信息学（2018）。https://api.semanticscholar.org/CorpusID: 65594699[3] Tai-Danae Bradley。2021年。作为拓扑算子导出的熵。熵 23, 9 2021 10 3390 23091195 ( )。doi: 。/e Ramprasad [4] Chinthekindi Shyam Burkule。2024年。热门分布式键值存储的比较分析。《国际科学与研究期刊》（*IJSR*）（2024）。 269835582 https://api.semanticscholar.org/ CorpusID:[5] P. Ceravolo 等。2024年。moduli：面向数据密集型工作流的解耦式数据管理架构。载于第*39*届 *ACM/SIGAPP*应用计算研讨会论文集（*SAC '24*）。计算机协会，纽约，美国。doi:10.1145/3643603. 3643607[6] Andr Lopes 等。2023年。图数据库系统的可扩展性与性能评估：Neo4j、JanusGraph、Memgraph、NebulaGraph 和 TigerGraph 的比较研究。载于*2023*年面向下一代计算的智能技术与系统国际会议论文集（*SmartTechCon*）。doi:10.1109/ SmartTechCon57526.2023.10391694[7] Aguilar V. 等。2023年。NoSQL 数据库建模与管理：系统性文献综述。《工程学院学报》32 09 2023 1 28 10 19053 01211129 32 65 2023 ( ), –。doi: . / .v .n . . 165119 [8] Chessa 等。... 以 2022 知识图谱丰富数据湖（短文）。载于*TEXT2KG/MK@ESWC*。 249010792 https://api.semanticscholar.org/CorpusID:[9] Claudio Caudai 等。2021年。功能基因组学中的人工智能应用。《计算与结构生物技术杂志》19（2021），6029–6040。doi:10.1016/j.csbj.2021.10.009[10] Elgendy 等。2014年。大数据分析：文献综述。《计算机科学讲义》8557（08 2014），214–227。doi:10.1007/ 978-3-319-08976-8 16_[11] Feng Ye 等。2023年。多模型数据库与多语言持久化的性能评估基准测试。《数据库管理杂志》(2023 10 4018 321756)。doi: . /jdm. [12] Gongsheng Yuan 等。... 半结构化数据和图数据映射到关系型数据的调查。*ACM* 通讯（2022）。doi:10. 1145/3567444[13] Johan Sandell等。2024。《ArangoDB、MySQL和Neo4j的性能比较分析：连接数据查询的实验研究》。（2024）。doi:10.48550/arXiv.2401.17482 预印本可在arXiv获取。[14] Khasawneh等。2020。《SQL、NewSQL和NoSQL数据库：比较综述》。载于*2020*年第*11*届国际信息与通信系统会议（*ICICS*）。013–021。doi:10.1109/ICICS49469. 2020.239513[15] Mishra等。2018。《面向文档的NoSQL数据库：实证研究》。载于数据科学与分析，Brajendra Panda、Sudeep Sharma和Nihar Ranjan Roy（编）。Springer Singapore，新加坡，126–136。[16] Maryam Abbasi等。2018。《Cassandra上的NoSQL可扩展性性能评估》。支持学习的信息系统与技术(2018 70051256)。[17] 2019 https://api.semanticscholar.org/CorpusID: Mert Erdemir等。《图数据模型与超图数据模型上的Neo4j查询性能比较》。载于第*11*届国际知识发现与信息检索会议（*KDIR*）论文集。 397 404 10 5220 0008214503970404 – 。doi: 。/ [18] 2024 Quoc-Bao-Huy等。《结合BERT、GraphSAGE和Transformer的稳健文本到Cypher（CoBGT）模型》。应用科学 14, 17 2024 7881 10 3390 14177881 ( ), 。doi: 。/ [19] 2025 app Simon Knollmeyer等。《文档GraphRAG：利用知识图谱增强检索增强生成，以回答制造领域中的文档问题》。电子学(2025)。https://api.semanticscholar.org/CorpusID: 278834060 [20] 2021 T. V. Zhidchenko等。《使用Neo4j图数据库设计能量负载系统》。*IOP*地球与环境科学会议系列659, 1 (2021), 012108。doi:10.1088/1755-1315/659/1/ 012108[21] Van Landuyt等。2023。《多模型NoSQL数据库与多语言持久化的比较性能评估》。载于第*38*届*ACM/SIGAPP*应用计算研讨会论文集（爱沙尼亚塔林）（*SAC '23*）。计算机协会，纽约，NY，美国，286–293。doi:10.1145/3555776.3577645[22] Van Landuyt等。2023。《多模型NoSQL数据库与多语言持久化的比较性能评估》。载于第*38*届*ACM/SIGAPP*应用计算研讨会论文集（*SAC*）。677–685。doi:10.1145/3555776.3577645[23] Kohei Hamaji和Yukikazu Nakamoto。2024。《一种使用自然语言处理的MongoDB文档重构支持系统》。软件 3, 2 2024 169 183 10 3390 3020010 ( ), –。doi: 。/ [24] 2021 software Aidan等，Hogan。《知识图谱》。计算机协会(2021 10 1145 3447772)。doi: 。/ G.et al. Huang。[25] 2023 《机器学习在材料合成与性质预测中的应用》。材料16, 17 (2023), 5977。doi:10. 3390/ma16175977[26] Aaesha T. Khanam。2024。《基于图的数据科学工具在揭示复杂网络关系中的作用》。国际科学与创新工程期刊(2024)。https://api. semanticscholar.org/CorpusID: 275769964 [27] 2024 Dipali Meher。《学习图数据库：Neo4j概述》。国际工程应用科学与技术期刊8 (01 2024 216 219 10 7753 1212 1007), –。doi: 。/IJ [28] 2025 CATR 。Xusheng Mei。《基于负载均衡的MongoDB数据库性能优化技术研究》。工程技术研究进展13, 1 (2025), 1062–1070。doi:10.56028/ aetr.13.1.1062.2025[29] Özcan等。2021。《面向人工智能应用的数据语义丰富化》。载于第五届端到端机器学习数据管理研讨会论文集（中国，线上）（*DEEM '21*）。计算机协会，纽约，NY，美国，文章4，7页。doi:10.1145/3462462.3468881

[30] Luiz Henrique Zambom Santana和Ronaldo dos Santos Mello。2022。《RDF数据持久化到NoSQL：综述与参考架构》。*IEEE*知识与数据工程汇刊 34(2022), 1370 1389 219465160 –。

[31] 2020 https://api.semanticscholar.org/CorpusID: Xin Wang和Weixue Chen。《知识图谱数据管理：模型、方法与系统》。载于*WISE*研讨会。https: //api.semanticscholar.org/CorpusID: 211080517

[32] 2020 Xin Wang 和 WeixueChen。《知识图谱数据管理：模型、方法与系统》。载于*WISE*研讨会。https: // api.semanticscholar.org/CorpusID:211080517