# 大语言模型时代的面向任务的数据集搜索再探讨：挑战、基准与解决方案

Zixin Wei 香港中文大学（深圳） 中国深圳
zi
xinwei1@link.cuhk.edu.cn

Yucan Guo*中国科学院计算技术研究所 中国北京
guoyucan23z@ict.ac.cn

Jinyang Li 香港大学 中国香港
jl0725@connect.hku.hk

Xiaolin Han 西北工业大学 中国西安
xiaolinh@nwpu.edu.cn

Xiaolong Jin*中国科学院计算技术研究所 中国北京
jinxiaolong@ict.ac.cn

Chenhao Ma†香港中文大学（深圳） 中国深圳
machenhao@cuhk.edu.cn

arXiv:2512.15363v1 [cs.DB] 17 Dec 2025

## 摘要

寻找合适的数据集是数据驱动研究的关键“第一步”，但这仍是一项巨大的挑战。研究人员通常需要根据高层次的任务描述搜索数据集。然而，由于用户意图模糊、任务到数据集的映射存在缺口以及实体歧义，现有搜索系统难以胜任这一任务。为应对这些挑战，我们推出了 KATS，这是一种全新的端到端系统，能够从非结构化科学文献中进行面向任务的数据集搜索。KATS 由两个核心组件构成，即离线知识库构建和在线查询处理。其先进的离线处理流程采用协作式多智能体框架进行信息抽取，自动构建高质量且可动态更新的任务-数据集知识图谱，从而弥补任务到数据集的映射缺口。为进一步应对实体歧义这一挑战，系统采用独特的基于语义的机制进行任务实体链接和数据集实体消歧。在在线检索方面，KATS 使用专门的混合查询引擎，将向量搜索与基于图的排序相结合，以生成高度相关的结果。此外，我们还推出了 CS-TDS，这是一个专为评估面向任务的数据集搜索系统而设计的基准测试套件，旨在弥补标准化评估方面的关键缺口。在该基准测试套件上进行的实验表明，无论在有效性还是效率方面，KATS 均显著优于最先进的检索增强生成框架，为下一代数据集发现系统提供了可靠的蓝图。

## 关键词

数据集搜索、面向任务的搜索、知识图谱 raph

**PVLDB 研究成果可用性：**源代码、数据和/或其他研究成果已发布于 https://github.com/starkersawz666/KATS。

*同时隶属于中国科学院大学计算机科学与技术学院。

†通讯作者。

## 1 引言

搜索适用的数据集是任何数据驱动的科学研究中基础却往往艰巨的一步。为回应这一迫切需求，数据集发现问题日益受到学术界和工业界的关注，推动了各类数据搜索系统的发展 [3, 5, 17]。然而，这些系统的主流范式仍然是基于关键词的检索，隐含地假设用户能够通过结构化元数据或关键词字段描述其数据需求。这种方法预设用户已将高层次的研究目标转化为精确的数据规范。

相比之下，近期的实证研究揭示了这一假设与现实行为之间存在显著偏差：对于 79% 的数据专业人员而言，寻找相关数据集是最常见且最关键的任务 [17]。这凸显了面向任务的数据集搜索的重要性，即根据自然语言任务描述检索数据集；这仍是一项长期存在且尚未解决的挑战 [7]。

我们确定了阻碍该领域进展的三个根本障碍：

- **用户意图模糊**。 用户的高层目标与搜索引擎通常要求的底层关键词查询之间存在显著的“语义鸿沟”。这种不匹配往往会导致结果不相关 [7]。
- **任务到数据集的映射与基准测试缺口**。即使系统能够理解用户意图，科学任务与其相关数据集之间仍缺乏明确的映射。此外，缺少标准化基准测试也阻碍了对面向任务的数据集搜索系统进行严格评估和比较研究。
- **实体歧义**。 文献中对数据集的引用方式不一致，例如使用缩写、全名或版本，这会导致结果分散且召回率较低 [7]。

幸运的是，近期的技术进步为应对这些挑战带来了新的机遇。大语言模型（LLM）展现出了出色的语义理解和推理能力 [39], , 因此非常适合解读复杂的任务描述。与此同时，检索增强

生成（RAG）范式 [23]提供了一种灵活的机制，可将大语言模型建立在外部知识源之上。

尽管取得了这些进展，现有的检索增强生成框架通常面向通用问答，而非任务导向的数据集发现。例如，HippoRAG 专注于多跳推理 [14], ，而 Raptor 的层次化设计可能会遗漏跨文档关联 [24]。更重要的是，在数据集检索场景中，实体歧义、任务-数据集映射和相关性排序等核心问题仍未得到充分解决。此外，大规模构建和运行检索增强生成系统仍然耗时且耗费资源。

为了解决这些问题，我们提出**知**识图谱**增**强**任**务导向数据集**搜**索（KATS），这是一个专为任务驱动的数据集发现而构建的系统。KATS 采用两阶段处理流程：（1）离线组件自动构建任务-数据集知识图谱（KG），并支持高效的增量更新；（2）在线混合查询引擎将大语言模型驱动的检索与基于图的推理相结合，以发现相关数据集。

重要的是，我们还推出了一个标准化的**基准**，用于面向任务的数据集搜索。该基准源自科学文献，与我们用于构建知识图谱的语料库类型相同。科学论文是这项工作的天然信息来源，因为它们通常会同时描述任务及其所使用的数据集。这种隐式对齐使我们能够大规模整理真实的任务-数据集对，为评估数据集发现系统提供可复现且具有代表性的框架。

我们的主要贡献总结如下：

- 我们基于经过整理的科学文献，构建了一个用于面向任务的数据集搜索的标准化基准测试套件。这填补了长期存在的评估空白，并支持对数据集发现方法进行系统评估。
- 我们设计了一种新颖的端到端处理流程，可自动构建任务-数据集知识图谱，并加入专用的实体消歧模块，以解决数据集命名歧义。
- 我们提出了一种混合查询引擎，将稠密检索与基于图的相关性排序相结合，从而提供准确且与任务相关的数据集推荐。
- 大量实验表明，KATS在有效性和计算效率方面均优于现有的基于大语言模型的方法。

本文其余部分安排如下：第 2 节回顾数据集搜索与检索增强生成系统的相关研究。第 3 节定义面向任务的数据集搜索问题，并介绍我们的基准。第 4 至 6 节详细介绍系统架构。第 7 节报告实验结果，随后第 9 节和第 10 节展开讨论并总结。

## 2 相关研究

### 2.1 数据集搜索系统

数据集搜索长期以来一直被视为学术界和工业界的一项基础性但颇具挑战的问题。近期一项调查显示，数据搜索最常见的目标是为给定任务找到初始数据集 [17]。然而，考虑到现有的数据质量和搜索方法，这一过程颇具挑战。现有搜索系统往往受元数据质量不佳的制约 [27]。

与此同时，用户意图与系统能力之间存在语义鸿沟，其中包括在可用数据集的背景下理解用户查询，以及匹配语义相关但句法不同的数据集 [30]。我们的工作立足于这一广泛的研究背景，旨在推动数据集发现范式从简单的关键词搜索向更智能、更以用户为中心的方法演进。

*2.1.1* 通用与网络规模系统。 在数据集搜索发展的早期阶段，研究者专注于构建网络规模的发现工具，其中最具代表性的是 Google Dataset Search [5]。该系统依赖数据所有者按照 Schema.org [34] 和 Croissant [1]等开放元数据标准发布结构化元数据。通过汇总这些元数据，系统能够为海量数据集建立索引。然而，这种方法的根本局限在于依赖高质量、经人工整理的元数据。当元数据缺失或质量较差时，搜索结果会显著下降。

*2.1.2* 基于语义的系统。 随着“数据湖”概念的普及， [4], 数据集发现面临着新的挑战。在大量异构数据集以原始格式存储且模式信息缺失或不一致的环境中，发现相关数据集变得更加复杂。为应对这些挑战，一系列先进的基于语义的系统应运而生。Aurum [32] 是具有代表性的语义系统。它构建了一个企业知识图谱，系统地捕获数据列之间的句法关系，使用户能够执行灵活的探索性查询。此外，D3L [4] 提出了一种基于距离的框架，结合多种类型的证据来发现数据湖中的相关表格。此外，Table Union Search [26]支持在开放数据环境中对表格数据进行联合搜索，从而提升相关表格发现的质量和搜索效率。同样，Auctus [6] 是一种支持数据增强查询的搜索引擎，允许用户查找可与给定表格合并的数据集。

大语言模型兴起后，集成大语言模型的系统展现了运用大语言模型进行数据集检索的强大能力。PNEUMA [3] 将大语言模型与传统信息检索技术相结合，能够根据自然语言查询，从大型集合中高效、准确地发现相关表格数据。LEDD [2] 利用大语言模型自动生成分层全局目录，并支持使用自然语言进行语义表搜索。此外，SemExplorer [37] 提供了专为定制化语义数据集搜索设计的用户界面，凸显了人机交互在解决复杂发现问题中的重要性。AUTOTUS [15] 结合了大语言模型的上下文理解能力与多阶段训练策略，实现了对表格间复杂关系的深度表示学习，显著提升了自动联合表搜索的准确性和鲁棒性。这一系列研究清楚地展现了范式转变：从依赖外部元数据转向深入理解数据内容。然而，这仍不足以充分满足用户寻找能够服务于特定“任务”的数据集的需求。

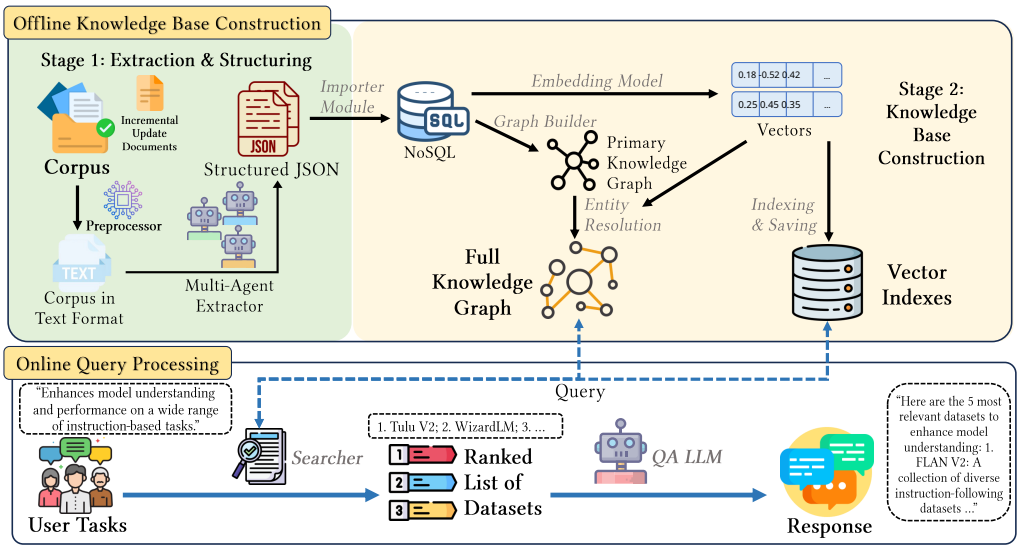



**图 1：KATS 系统的整体架构。**

*2.1.3* 任务驱动的数据集搜索。任务驱动范式将成为数据集搜索领域的一项关键转变，其动因源自数据专业人员在现实世界中的工作实践。近期一项针对数据专业人员的调查 [17] 将任务驱动搜索列为下一代数据集搜索系统的四项关键需求之一。这凸显了直接源于用户实践的一项迫切需求：用户往往不清楚所需数据集的名称，而是希望找到适合完成特定任务的数据集。

近期研究已开始探索这种由任务驱动的范式。例如，METAM [12] 提出了一种目标导向框架，其中“任务”是一个可计算的机器学习（ML）模型。系统会不断扩充数据集，以提升定量效用分数，例如 F1 分数。类似地，Mileena [16]也将任务定义为 ML 模型，但通过使用半环草图，着重解决搜索延迟和数据隐私这两项关键挑战。

尽管这些系统功能强大，也解决了该问题的重要方面，但它们对“任务”的定义依赖于形式化且可量化的函数。相比之下，我们的工作着眼于一种更常见的场景：用户以开放式自然语言描述任务。这种表述需要一种根本不同的方法，着重深入理解用户意图的语义，以及这种意图与非结构化文本中知识之间的关系。解决这一特定挑战——基于自然语言的任务驱动搜索——是我们开展这项工作的主要动机。

## 2.2 基于知识图谱的检索增强生成

检索增强生成的出现为知识密集型任务开辟了新的领域。近期研究已超越简单的基于向量的检索，提出了更复杂的架构，使得能够构建知识语料库以实现更高效的检索。因此，这些方法适用于从非结构化科学文献中进行数据集搜索。我们将本研究置于其中若干最先进框架的背景下，并在实验中将它们用作强基线方法。

这些先进系统代表了不同的知识结构化策略。GraphRAG [11] 使用大语言模型从源文档中构建基于实体的知识图谱，并识别社区，生成抽象式摘要，以回答广泛的问题。HippoRAG 系列研究提出了一种受神经生物学启发的框架，旨在增强检索增强生成的推理能力。最初的 HippoRAG [14] 提出了核心思想：在由大语言模型构建的知识图谱上使用个性化PageRank，以实现多跳检索。随后，其继任者 HippoRAG 2 [20] 在此基础上进一步改进，以解决最初以实体为中心的方法的若干局限。具体而言，HippoRAG 2 引入了段落节点，以及更精细的查询到三元组匹配机制，以实现更深入的上下文理解。此外，它还整合了识别记忆模块，以提高检索的准确性和效率。RAPTOR [33] 通过递归地聚类和总结文本块来构建层级结构，使系统能够在不同抽象层级检索信息，从而针对长文档进行通用问答。

尽管这些系统是功能强大的通用框架，但它们并未针对从科学文献中进行数据集发现这一任务所面临的独特挑战进行专门优化。相比之下，我们提出的 KATS 系统是一个专为这一特定任务定制的端到端流水线。该系统融入了解决实体歧义、数据集排序等专门机制，而这些并不是更通用的检索增强生成架构所关注的重点。

## 3 问题定义

本节对我们的核心问题——面向任务的数据集搜索问题——进行了正式定义。随后，我们介绍了新基准测试套件的构建过程及其特征。该套件旨在严格评估系统在这一特定任务上的表现。

## 3.1 面向任务的数据集搜索问题

设 $C = \{c_1、c_2、\cdots c_n\}$ 为一个由非结构化学术文献构成的语料库。给定用户以自然语言提出的任务查询 $T$，面向任务的数据集搜索系统的目标是返回数据集排序列表 $D = \{d_1、d_2、\cdots d_k\}$，其中每个数据集$d_i$都对应语料库 $C$ 中唯一的数据集实体，并且与解决 $T$ 所描述的任务相关。系统的目标是最大限度地提高所返回数据集列表 $D$ 的相关性。数据集与任务查询的相关性根据我们在基准测试套件中制定的真实标注标准进行定义（见第 3.2 节）。

## 3.2 基准测试套件

虽然已有一些用于问答（例如 Hot- PotQA [38]）或通用文档检索（例如 BEIR [35]）的基准测试，但目前仍缺乏专门用于评估在大型科学文献语料库中进行面向任务的数据集搜索的成熟基准。为实现严格且可复现的评估，我们构建了一个新的基准测试套件，称为计算机科学任务数据集搜索（CS-TDS）。

*3.2.1* 语料库整理。我们的套件包含两个基准测试，它们基于不同规模的语料库构建，用于测试系统性能和可扩展性；详细统计数据汇总于表 1。第一个基准 CS-TDS$_M$由包含 628 篇计算机科学论文的语料库构建，第二个规模更大的基准 CS-TDS$_L$ 则基于 2101 篇论文构建。这些论文由网络爬虫从计算机科学会议（例如 NeurIPS）获取。

**表 1：CS-TDS 基准测试套件的统计信息。**

| 统计项 | CS-TDS$_M$ | CS-TDS$_L$ |
|---|---|---|
| 语料库中的论文数量 | 628 | 2，101 |
| # 任务查询 | 47 | 204 |
| # 数据集 | ~ 1，779 | ~ 7，525 |
| # 唯一数据集 | ~ 1，217 | ~ 4，180 |
| 年份跨度 | 2023 | 2024 |

*3.2.2* 查询生成。 这些查询由大语言模型根据真实论文生成，采用自然语言描述任务，并经过人工核验。CS-TDS$_M$选取了 47 个查询，CS-TDS$_L$选取了 204 个。为确保评估测试的是系统的泛化能力，而非仅仅检索源文档的能力，查询及真实答案所依据的论文不会纳入搜索语料库。以下是我们基准中的一个代表性示例：

**基准查询示例**

**任务查询：** “评估 *HSIVI-SM* 在保持样本质量的同时加速扩散模型采样的能力，并通过 *Fréchet Inception Distance*（*FID*）进行测量。”

**真实标注数据集：** *“CIFAR-10”*，其中包含 60， 000 32 ×32 张分属 10 个不同类别的彩色图像 [22]。

*3.2.3* 真实标注。 对于每个查询，标准答案设为论文中实际使用的相应数据集。为避免“假阴性”，除原始答案外，其他合理答案也视为正确答案。

形式化而言，如果数据集 $d_i$ 在真实标注过程中满足以下任一条件，则被视为与任务查询 $T$相关：

- 它与来源论文中用于 $T$的数据集**完全相同**。
- 它是该数据集的**变体或别名**（例如，MIMIC-III 与重症监护医学信息集市 III）。
- 它是**功能等效的替代品**，通常来自同一数据集系列或具有相同的任务目标（例如，用于大规模图文预训练的 LAION-5B 与 LAION-400M）。

这些标准和相应的标注均经过人工验证，并用于在评估期间确定二元相关性标签。系统根据其针对每个查询检索此类相关数据集的效果进行评分。

## 4 系统概述

本节将详细介绍我们提出的 KATS 的设计。首先，我们介绍整体架构以及各组件的功能。随后，我们正式定义知识图谱的数据模型，这是系统所管理的核心数据结构。

## 4.1 整体架构

如图 1 所示，KATS 的架构分为两个主要阶段：用于知识库构建的离线阶段，以及用于查询处理的在线阶段。

*4.1.1* 离线知识库构建。 离线阶段始于阶段 1：提取与结构化。在此过程中，输入论文会被转换为结构化格式，以便后续进行知识库构建。预处理器模块将异构源文档（主要为 PDF 格式）规范化为纯文本表示。随后，下游抽取器模块使用大语言模型，将任务和数据集信息提取为 JSON 格式。

随后，结构化信息会被传递至阶段 2：知识库构建。在将结构化信息导入数据库后，该阶段包含两个相互关联的工作流。向量工作流为任务和数据集生成稠密向量表示，并将其存储在向量索引中。同时，在图工作流中，图构建器模块填充初始知识图谱。随后，实体消歧模块利用此前生成的向量嵌入进一步完善知识图谱。

*4.1.2* 在线查询处理。在线阶段旨在支持实时用户交互。当用户提交自然语言任务查询时，混合检索器会与离线知识库交互。它会查询向量索引以获取语义相似度，并查询知识图谱以获取任务-任务和任务-数据集关系信息。搜索器结合这些信号，生成与用户任务最相关的数据集排序列表。随后，该列表会传递给下游问答大语言模型，由其重新排序结果以提高精确率，并生成面向用户的最终自然语言回答。

### 4.2 任务-数据集知识图谱

KATS 管理的核心数据结构是任务-数据集知识图谱。该图形式化定义为 $G=(V,E)$，其中 $V$ 表示节点（顶点）集合，$E$ 表示无向边集合。

**表 2：符号及其含义。**

| 符号 | 含义 |
|---|---|
| $G=(V,\ E,\ W)$ | 顶点集为 $V$ =的无向加权知识图谱 $C\cup D\cup T$，边集为 $E=E_{C,\ D}\cup E_{D,\ T}\cup E_{T,\ T}$，以及权重函数 $W$ . |
| $C$ | 文档节点集合。 |
| $D$ | 数据集节点集合。 |
| $T$ | 任务节点集合。 |
| $E_{C,D}$ | 边集合（$c,\ d$），其中 $c\in C, d\in D.$ |
| $E_{D,T}$ | 边集合（$d,\ t$），其中 $d\in D, t\in T$。 |
| $E_{T,T}$ | 边集合（$t_1,\ t_2$），其中 $t_1,\ t_2\in T$。 |

顶点集合 $V$ 由三种不同类型的节点组成：

- **文档节点**：表示输入中的单个文档（论文）。每个节点都存储文档的元数据，例如标题和源文件路径。
- **数据集节点**：表示文档中使用的数据集实体。每个节点都包含属性，例如数据集的名称和简要描述。
- **任务节点**：表示文档中描述的、数据集所用于的特定任务。每个节点都包含属性，例如对任务的文本描述。

边的集合 $E$ 捕捉了这些节点之间的关系：

- **文档-数据集边**：一条边（$c,\ d$），其中 $c\in C, d\in D$ $c$ $d$ 表示文档包含数据集。
- **数据集-任务边**：一条边（$d,\ t$），其中 $d\in D, t\in T$ 表示数据集 $d$ 用于完成任务 $t$。
- **任务-任务边**：一条边（$t_1,\ t_2$），其中 $t_1,\ t_2\in T$ 及其权重 $w$ 表示任务 $t_1$ 和 $t_2$ 在语义上相似，其余弦相似度为 $w$。

## 5 知识库构建

本节将介绍构建任务-数据集知识库的离线流程详情。尽管当代基于图的 RAG 系统也会从文本中构建知识图谱，但其方法通常针对通用摘要进行定制。然而，对于精确的面向任务的数据集搜索，这种方法可能并不理想，因为任务上下文等关键细节可能会丢失。此外，这些通用框架通常缺乏解决实体歧义问题所需的专用机制。为克服这些局限，我们的处理流程专门设计用于精确提取、整理和查询数据集实体与任务实体。该流程从原始文档开始，并在一个

一系列步骤。此外，为确保知识库的长期可行性和可扩展性，我们设计了该系统，使其能够高效地进行增量更新，从而保持最新状态而无需耗费大量成本定期重建。以下小节将进一步讨论每个步骤。图 2 展示了本体以及任务-数据集知识图谱的示例。

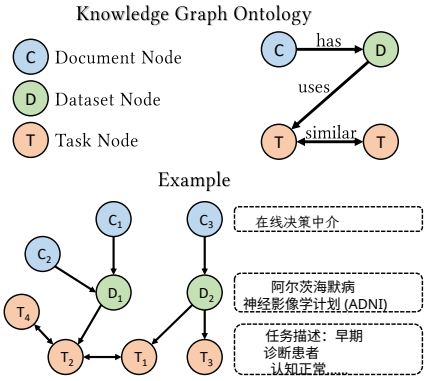



**图 2：本体及任务-数据集知识图谱示例。**

### 5.1 文档标准化与缓存

处理流程的第一步是收集并标准化源文档。此步骤的主要目标是将各种格式的文档语料库转换为纯文本。主要挑战在于处理 PDF 格式论文的复杂性，以确保大语言模型能够在后续步骤中准确提取信息。

在当前实现中，我们使用 pypdf [31] 库进行直接文本提取，效率很高。由于我们的目标实体——数据集及其对应的任务——主要出现在论文的叙述性正文中，因此图和表格等非文本元素在我们的提取策略中并非关键。因此，这种方法足以可靠且有效地提取目标实体。为了应对更复杂的情况，例如大量文本以图像或复杂的多栏布局呈现时，我们可以考虑使用光学字符识别 (OCR) 或智能文档解析服务（例如 LlamaParse [25]），但这类服务的计算成本较高。此步骤完全由独立模块完成。因此，可以根据源文档的具体情况替换相应方法，以提高效率并节省成本。

为优化性能，尤其是在处理大型静态语料库时，我们采用了缓存策略。处理每个文件之前，系统会计算文件内容的唯一哈希值，作为其“指纹”。在后续运行中，如果遇到指纹相同的文件，系统便会直接从缓存中读取其内容，从而避免重复计算带来的开销。在后续的信息抽取步骤中，我们也采用了相同的缓存方法。

### 5.2 信息抽取

将文档规范化为纯文本后，抽取器模块负责识别并提取有关数据集及其相关任务的结构化信息。一些当代方法，如 GraphRAG [11], ，会执行一次性、

通用实体抽取。相反，我们专门针对面向任务的数据集发现这一目标，设计了一个协作式多智能体框架。该框架将复杂的抽取挑战分解为一系列子任务，每个子任务由一个专用智能体处理。这种设计不仅提高了所提取信息的准确性和相关性，还通过避免对无关文档进行深入处理来提升效率。

*5.2.1* 初步筛选智能体。我们框架中的第一个智能体充当高效的把关者。它的主要职责是快速识别可能提及数据集的文档，从而避免对无关内容进行耗费较大的大语言模型调用。对于每篇文档，该智能体都会使用简洁的指令模板向大语言模型发出查询，以获取一个简单的关于数据集相关性的布尔响应。只有被判定为相关的文档才会传递给后续智能体。

*5.2.2* 分析智能体。分析智能体专注于深入的内容分析。它利用详细的指令模板引导大语言模型仔细阅读文本，以帮助识别所有数据集提及及其相关任务。为确保机器可读性，该智能体必须按照预定义的结构化 JSON 格式生成输出。此格式由一组对象构成，每个对象表示一种独特的数据集—任务关系，并包含相关信息。表 3 展示了论文中一段输入文本片段 [18] 以及该智能体生成的相应结构化 JSON 对象的具体示例。

*5.2.3* 丰富化智能体。 最后，将结构化 JSON 对象传递给丰富化智能体。该智能体的唯一目的是增强所提取实体的语义元数据。对于每条任务描述，智能体都会使用单独的指令模板，生成一组相关关键词，用于概括任务的核心概念。这些关键词有助于后续阶段的图谱整理。

## 5.3 语义表示学习

提取结构化文本信息后，下一步是将这些文本转换为有意义的向量表示。此过程会将文本描述映射到高维向量空间中。这些向量嵌入是我们系统的基础，既用于图谱构建中的相似度定量分析，也用于在线查询中的语义搜索。

在我们的系统中，我们为两类关键实体生成嵌入：任务和数据集。对于每个任务节点，我们嵌入其任务描述。对于每个数据集节点，我们将其标题和描述拼接起来，构成全面的表示。在我们的实现中，我们通过应用程序接口使用 OpenAI 的 text-embedding-3-small 模型 [29] 。我们选择该模型，是因为它在嵌入质量、成本效益和集成便利性之间实现了出色的平衡。

生成的嵌入向量 $\{e_{task}\}$ 和 $\{e_{dataset}\}$ 随后会被持久化并建立索引，以实现高效检索。具体而言，我们利用 Faiss 库 [10] 从稠密向量中构建索引。Faiss 支持高效的近似最近邻搜索，使我们的系统能够实时从大型集合中找到语义上最相似的项。该索引步骤确保在线语义搜索和离线图谱构建过程都能以低延迟执行相似性查找。

## 5.4 实体消歧与链接

从抽取阶段构建的初始知识图谱虽然较为全面，但仍然包含自然语言固有的歧义和冗余，尤其是在任务节点和数据集节点方面：

- 语义相同的任务可能会用不同的措辞来描述，例如“图像分类”和“对图像进行分类”。
- 同一数据集可能有不同的名称或缩写，例如“COCO”和“MS-COCO”。

为了提升知识库的质量和可靠性，我们执行图谱整理阶段，其中包含两个主要流程：任务链接和数据集消歧。

*5.4.1* 任务链接。 为了解决任务描述的歧义，我们的系统会识别并连接语义相似但措辞不同的任务节点。为实现这一目标，TaskMerger 模块使用向量嵌入 $\{e_{task}\}$。此过程使用 Faiss 查找向量余弦相似度较高的任务对。如果两个任务 $t_1$ 和 $t_2$ 之间的相似度得分超过预设阈值 $\theta_d$，或者关键词重叠超过阈值 $\theta_k$，我们就认为它们在语义上密切相关。

识别出此类任务对后，系统会在知识图谱中相应的任务节点之间添加一条加权边。该边的权重设为计算得到的相似度分数，用以表示两者关系的强弱。这些相似度边对于在线阶段基于知识图谱的任务扩展至关重要。进行查询时，系统能够通过这些边，将搜索范围扩展到更多相关任务。

*5.4.2* 数据集消歧。 数据集消歧旨在解决实体冗余问题，即同一数据集可能以不同名称或缩写出现。对数据集节点进行全对比较的复杂度为 $O(n^2)$，计算开销难以承受。为解决这一问题，我们设计了一种高效且有效的三阶段方法。

第一阶段着重于高效生成可能需要合并的候选对。我们没有采用蛮力比较，而是使用 Faiss 索引。对于每个数据集，系统都会执行近似最近邻搜索，以检索一小组语义上最相似的数据集。此过程将搜索空间从所有可能的配对缩减为数量可控且可能性较高的候选项。初始时，每个数据集都作为一个独立节点存储在不相交集数据结构中。

在第二阶段，系统通过标题规范化执行初始匹配。具体而言，系统会对大小写、空格等进行标准化处理。此外，系统还维护一个字典，用于记录数据集的别名。经过这些处理后，具有相同规范化标题的数据集会被合并到不相交集中。

上述方法可以部分缓解同一数据集使用不同名称的问题，例如“Image Net”和“imagenet”。对于更复杂的情况，大语言模型会在第三阶段充当专家判定器，执行严格的验证流程。提示词会提供两个数据集的名称和描述，并要求模型以二元答案“True”或“False”判断它们是否指向完全相同的实体。如果大语言模型给出肯定判断，系统便会合并知识图谱中的两个节点。

**表 3：基于大语言模型的信息抽取流程示例。**

| 组件 | 内容 |
|---|---|
| **输入文本片段** | 在阿尔茨海默病领域，任务是对阿尔茨海默病神经影像学中的患者进行早期诊断<br>将该计划中的研究对象分为认知正常、轻度认知障碍或有痴呆风险者。 |
| **生成的 JSON 输出** | {"dataset_name": "阿尔茨海默病神经影像学倡议 (ADNI)", "task_description": "早期诊断<br>患者是否认知正常、轻度认知障碍或有痴呆风险。", "task_keywords": ["早期诊断",<br>"痴呆", "患者评估"]} |

同时，无论判断结果如何，都会将其存储在缓存中，以避免重复调用大语言模型。

这种混合式三阶段方法将向量搜索的可扩展性与大语言模型的推理能力相结合。因此，我们的系统为数据集实体消歧问题提供了一种稳健而高效的解决方案。

## 5.5 增量知识库更新

为了解决静态知识库的局限，并确保系统始终保持最新，我们设计了一套高效的增量更新处理流程。该机制使 KATS 能够将新抽取的知识整合到现有知识图谱和向量索引中，而无需对整个知识库进行全面且耗费大量成本的重建。

处理流程首先通过维护一份包含所有已处理文件的清单来识别新文档。每个文件都由其独特的内容指纹标识，该指纹通过从文件中提取文本并应用 SHA256 算法生成。将新文档添加到语料库时，系统会计算其指纹，并与清单进行比较，从而建立仅包含新文件的队列。每个新文档都会经过预处理和多智能体抽取流水线，随后将抽取出的结构化数据导入数据库。此过程本质上是增量式的，因为新条目只需添加到各自的表格中，不会影响现有数据。

为了更新语义搜索索引，系统会为这些新条目生成向量嵌入，并将其追加到现有的 Faiss 索引中。之后，将包括文档、数据集、任务及其基本关系在内的新节点添加到现有知识图谱中。随后，实体链接和消歧模块将以增量模式运行。对于每个新任务节点和数据集节点，系统会计算其与整个索引的相似度，以建立新的“相似_任务”边，并识别潜在的数据集合并。这种方法可确保新实体正确整合到现有图结构中，同时保留此前计算出的所有关系。整个处理流程确保 KATS 不只是一个静态存储库，而是一个能够随时间推移不断演进其知识的动态系统。

此外，为了利用科学文献之外更广泛的数据信息，增量更新功能还支持整合缺少明确任务描述的数据集。通过嵌入新的描述，并在现有知识图谱中找出最相似的数据集，KATS 可以自动将新条目关联到相关任务，从而利用更广泛来源的信息来丰富知识库。

# 6 查询处理

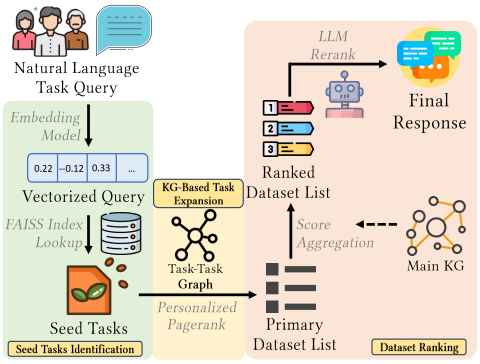



**图 3：KATS 系统的查询处理流水线。**

图 3 展示了在线查询处理流程的完整工作流。该流程由用户提交自然语言任务查询触发。我们的方法采用多阶段漏斗式设计，逐步将搜索空间从宽泛的语义匹配细化至精细的基于图的排序。

## 6.1 种子任务识别

在线查询处理流水线始于种子任务识别阶段。该阶段的主要目标是高效弥合用户自然语言查询与知识图谱中结构化实体之间的语义鸿沟。为此，系统会识别出一组初始的“种子”任务，这些任务在语义上与用户的查询最为相关。

该阶段包括两个主要步骤。首先，用户的任务查询 $T$ 被转换为高维向量表示 **t**。此过程使用与离线知识库构建阶段相同的预训练嵌入模型完成。这确保查询和知识库处于同一语义空间中，从而能够进行有意义的比较。接下来，使用查询向量 **t** 在预先构建的任务嵌入 FAISS 索引中执行近似最近邻（ANN）搜索。该搜索能够高效地从知识图谱中检索排名前 k 的任务节点，我们将其称为“种子任务”（$T_{seed}$）。$T_{seed}$ 是后续基于知识图谱的任务扩展的关键起点，可用于发现范围更广的相关信息。

## 6.2 基于知识图谱的任务扩展

初始种子任务集合 $T_{seed}$ 虽然在语义上接近查询，但可能并不全面。基于知识图谱的任务扩展阶段利用任务间相似度图中的关系信息，发现范围更广的相关任务。我们采用 Personalized PageRank（PPR）算法 [19] 进行扩展。PPR 能够计算相对于特定“源”节点集合的节点相关性。在我们的场景中，该算法在任务间相似度图上运行。此前识别出的种子任务 $T_{seed}$ 被用作 PPR 的个性化向量，这意味着“随机游走”过程会偏向于跳转回这些种子节点。因此，初始候选项的相关性分数会有效地扩散到整个图中。参数 $\alpha$ 用于平衡对图结构的探索与初始种子集合的影响。

此阶段的输出是知识图谱中每个任务节点的全局感知相关性分数。该分数既反映任务的直接相似性，也体现其在更广泛的相关任务网络中的互联程度。经过增强的任务分数集将作为最终数据集聚合与排序阶段的输入。

## 6.3 候选数据集聚合与排序

在最后阶段，系统聚合任务级相关性分数，生成最终的数据集排序列表。此过程包含两个关键步骤：首先基于图结构进行初始分数聚合，然后使用大语言模型进行最终重新排序，以提高精确率。

为聚合候选数据集，系统会识别主知识图谱中与高分任务节点相连的所有数据集节点。为计算每个唯一数据集的相关性分数，我们采用一种聚合策略。数据集的分数定义为所有与其相连的任务节点的相关性分数中的最大值：

$$Score(d) = \max_{t \in \text{LinkedTasks}(d)} \{Score(t)\}$$

根据这些聚合分数，我们将所有候选数据集按分数降序排列，生成初始排序列表。

为进一步提高排序结果中靠前项的精确率，我们引入最后一个基于大语言模型的重新排序步骤。系统从初始列表中选取排名前 $k_{rerank}$ 的数据集。对于每个候选项，系统都会从知识库中检索其详细描述。随后，系统将这些候选描述与原始用户查询一起整理成专门的指令，并要求大语言模型充当专家评估器，根据候选项与用户任务的直接相关性对其重新排序。该步骤利用大语言模型出色的理解和推理能力，为用户提供高度精炼且精准的最终结果。

# 7 实验评估

## 7.1 实验设置

*7.1.1* 数据集与基准。我们的实验基于基准测试套件 CS-TDS 开展。为确保评估公平且严谨，我们人工比较每个查询的groundtruth与output，以减少假阴性。除了原始数据集，任何属于同一系列（例如，不同规模的数据集）且用途相同的其他数据集也被标记为正确答案。

*7.1.2* 基线系统。 为评估 KATS 的有效性，我们将其与一系列强大的基线方法进行比较。这些方法包括一种成熟的稠密检索方法，以及若干专为复杂信息检索和问答设计的先进检索增强生成系统：

- **VanillaRAG**：作为成熟的语义基线，我们采用稠密段落检索（DPR） [21]。该方法首先将源文档拆分为更小的文本段落。然后，使用预训练的双编码器模型为每个文本段落和输入查询生成稠密向量表示。检索通过执行最大内积搜索来完成，以找出语义上最相关的文本段落。

- **GraphRAG**：GraphRAG [11] 使用大语言模型从源文本中构建知识图谱，其中节点表示提取出的实体，边表示它们之间的关系。检索时，通过解析用户的查询，在图中识别相关社区或路径。我们将预处理后的文本语料库输入其图构建引擎，以使其适用于我们的任务。
- **HippoRAG 2**：HippoRAG 2 [20] 使用开放信息抽取（OpenIE）来识别文档中的事实和实体，并将它们与图结构中的原始文本段落关联起来。检索过程在图上应用个性化 PageRank 算法，以找出最相关的信息。
- **Raptor**：Raptor [33] 递归地对文本块进行聚类，并为每个聚类生成摘要，以构建树状索引。这种分层结构使其能够在多个抽象层级上将查询与文本进行匹配。在我们的实验中，Raptor 的实现来自 DIGIMON [40]。

*7.1.3* 评估指标。 我们从有效性和效率两个方面评估所有系统的性能。

**有效性指标**

- **命中率@$k$**：在前 $k$ 个排序结果中找到正确数据集的查询所占比例。我们报告 $k$ 取 1、3、5、10 时的结果。
- **精确匹配（EM）**：对于排名第一的结果，如果检索到的数据集的规范名称与真实答案完全匹配，则该指标为 1；否则为 0。
- **F1 分数**：同样针对排名第一的结果，我们计算预测的数据集名称与真实名称之间的词元级 F1 分数，以衡量部分匹配情况。

**效率与成本指标**

- **离线构建时间**：执行整个离线知识库构建流程所需的实际运行总时间。
- **存储占用**：所有生成的产物所占用的总磁盘空间，可能包括知识图谱、向量索引和任何缓存。
- **构建令牌成本**：离线构建阶段所有大语言模型 API 调用消耗的输入和输出令牌总数。
- **单次查询时间成本**：在线阶段单次查询的平均实际运行时间。
- **单次查询令牌成本**：在线阶段中，针对一次查询的大语言模型 API 调用所消耗的平均令牌数。

**表 4：CS-TDS$_M$ 和 CS-TDS$_L$上的详细有效性比较。对于每项指标，最佳结果以粗体显示，次佳结果带下划线。**

| 系统 | 命中率@1 | | 命中率@3 | | 命中率@5 | | 命中率@10 | | 精确匹配（Top-1） | | F1（Top-1） | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $C_M$ | $C_L$ | $C_M$ | $C_L$ | $C_M$ | $C_L$ | $C_M$ | $C_L$ | $C_M$ | $C_L$ | $C_M$ | $C_L$ |
| **KATS（我们的方法）** | **0.447** | **0.446** | **0.638** | **0.672** | **0.723** | **0.730** | **0.787** | **0.804** | 0.208 | **0.206** | **0.350** | **0.312** |
| GraphRAG | 0.340 | 0.309 | 0.468 | 0.475 | 0.596 | 0.534 | 0.617 | 0.601 | 0.064 | 0.034 | 0.260 | 0.192 |
| HippoRAG 2 | 0.298 | 0.304 | 0.426 | 0.515 | 0.447 | 0.608 | 0.468 | 0.662 | 0.170 | 0.093 | 0.246 | 0.192 |
| Raptor | 0.340 | 0.284 | 0.404 | 0.431 | 0.511 | 0.520 | 0.638 | 0.583 | **0.213** | 0.113 | 0.256 | 0.205 |
| VanillaRAG | 0.255 | 0.221 | 0.404 | 0.358 | 0.426 | 0.422 | 0.468 | 0.471 | 0.149 | 0.064 | 0.206 | 0.149 |

*7.1.4* 实现细节。 在所有实验中，我们的 KATS 系统均配置为使用 OpenAI 的 text-embedding-3-small [29] 作为嵌入模型。所有抽取和判断任务使用的大语言模型均为 gpt-4o-mini [28]。KATS 搜索器的关键超参数如下：

• **初始候选数量**（$k$）：为每个查询从 FAISS 索引中检索的初始候选任务数量设置为 $k = 2$。此参数控制初始语义搜索的范围。

• **PageRank 阻尼系数**（$\alpha$）：个性化 PageRank 算法的 alpha 参数用于平衡初始查询向量与图结构的影响，其值设置为 $\alpha = 0.85$。

所有实验均在一台服务器上进行，该服务器配备两颗 Intel Xeon Gold 6330 CPU、一块显存为 24GB 的 NVIDIA RTX A5000 GPU，以及 1TB 内存。系统运行于 Linux 发行版，CUDA 版本为 12.8。

## 7.2 端到端有效性

本节将介绍所提出的 KATS 系统相较于基线方法的端到端有效性评估。图 4 展示了以命中率@5 为指标的主要对比结果。结果清楚地表明，在 CS-TDS$_M$ 和 CS-TDS$_L$ 上，KATS 始终显著优于所有基线方法。

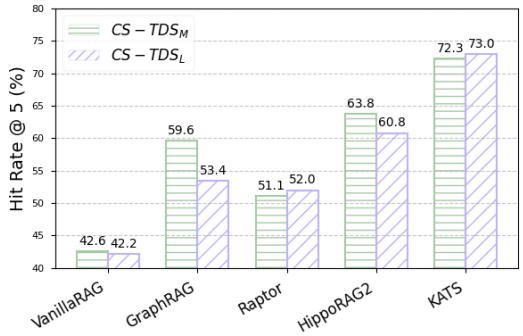



**图 4：CS-TDS$_M$ 和 CS-TDS$_L$ 上以命中率@5 为指标的端到端有效性对比。**

例如，在规模较大的数据集 CS-TDS$_L$ 上，KATS 的命中率@5 达到 73.0%。与最强基线 Hip-poRAG 2（60.8%）相比，这一结果提升了近 20 个百分点。与.....相比时，性能差距更为显著 VanillaRAG 基线，这凸显了我们基于知识图谱的混合搜索机制的有效性。在 CS-TDS$_M$ 数据集上，我们观察到类似趋势，证实了我们的方法在不同语料库规模下具有鲁棒性。

KATS 显著的性能优势可归因于若干关键设计选择。首先，与通用框架不同，我们的处理流程构建了高度专用的任务-数据集知识图谱，仅关注与目标相关的核心实体和关系。其次，我们创新的基于大语言模型的数据集实体消歧流程，通过合并名称不同但语义相同的节点，有效清理了图中的重复内容。这一步整理对于整合知识和提高召回率至关重要。最重要的是，我们的混合搜索机制利用 PPR算法进行复杂的任务扩展。这使系统能够在更广泛的语境中理解用户查询，并超越初始的关键词匹配或向量匹配，发现与更广泛语义相关任务关联的数据集。这直接促成了其相较于更直接检索方法的优越性能。

我们方法的优势并不局限于命中率@5这一项指标。表 4 对所有有效性指标进行了更全面、详细的比较。这表明 KATS 能够准确识别正确数据集的规范名称。即使搜索结果并非完全精确，其内容仍然高度准确。总而言之，高召回率与高精确率相结合，从多个方面有力地证明了 KATS 框架的端到端有效性。

## 7.3 效率与成本分析

除了有效性之外，数据集搜索系统的实际可用性还取决于其效率和运营成本。本节从三个关键维度评估 KATS 和基线方法：时间成本、应用程序接口成本（令牌消耗）和存储成本。我们的分析基于 CS-TDS$_M$ 基准开展。图 5 展示了有效性与效率之间权衡的整体情况。

我们首先分析离线构建和在线检索的时间成本，结果汇总于表 5。对于离线构建时间，更简单的 VanillaRAG 基线如预期般速度最快，这得益于其直接的索引流程。我们提出的 KATS 系统在 CS-TDS$_M$ 语料库上构建知识库需要 5.2 小时。虽然这比 VanillaRAG

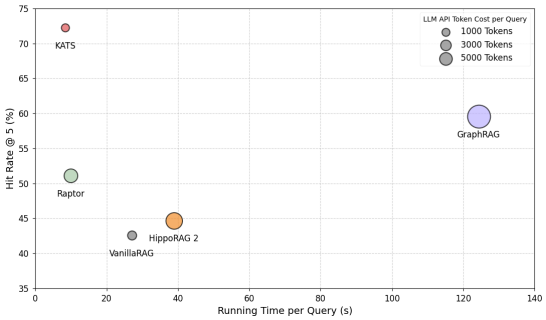



**图 5：KATS 中有效性与效率之间的权衡。**

**表 5：CS-TDS$_M$ 上离线构建（B.）和在线检索（R.）期间的时间、应用程序接口和存储成本比较。**

| 系统 | Time | | API | | 存储 (MB) |
|---|---|---|---|---|---|
| | B.（小时） | R.（秒） | B.（百万） | R. | |
| **KATS（我们的方法）** | 5.2 | 8.4 | 21 | 1，041 | 255 |
| GraphRAG | 99.4 | 124.3 | 151 | 20，011 | 8,228 |
| HippoRAG 2 | 233.7 | 38.9 | 57 | 9，670 | 10，693 |
| Raptor | 29.8 | 10.0 | 30 | 6，067 | 1，017 |
| VanillaRAG | 2.9 | 27.1 | 0 | 1,723 | 418 |

基线更长，但其效率显著高于其他复杂的、基于图的框架，例如 Raptor（29.8 小时）。这表明，我们的专用处理流程在所构建知识图谱的丰富性与其构建计算成本之间实现了良好平衡。鉴于该构建流程能够带来更优的在线性能，其成本是值得的。对于决定用户体验的在线检索时间，KATS 专为高效率而设计。得益于预先计算的图结构和快速的 FAISS 索引，我们的系统能够实现较低的查询延迟，每次查询约为 8.4 秒。这确保了响应迅速且具有交互性的搜索体验，而这正是任何实用数据集发现工具的关键要求。

除了时间之外，大语言模型 API 的令牌消耗也是实际部署中的关键因素。表 5 详细列出了离线构建阶段和在线查询阶段的令牌成本。在构建阶段，我们的多智能体提取模块展现出很高的成本效益。初步筛选智能体会剔除无关文档，大幅减少对功能更强大的分析智能体和丰富化智能体进行高成本调用的次数。同时，我们的混合数据集解析方法减少了重复的大语言模型调用。因此，与可能对整个语料库进行深度处理的基线方法相比，我们的总令牌消耗大幅降低。在查询阶段，主要的应用程序接口成本来自最后的大语言模型重排序步骤。由于此步骤只处理少量排名靠前的候选项，单次查询令牌成本极低。这种设计确保用户能够受益于基于大语言模型的重排序所带来的高精确率，同时无需承担过高的运行成本。

最后，我们分析了生成产物的存储占用，如表 5 所示。结果表明，我们的 KATS 系统保持了适中的存储占用。这一出人意料的结果源于架构上的根本差异。标准的检索增强生成方法必须为整个语料库中的每个文本块创建向量嵌入。相比之下，我们的 KATS 流水线会先智能地识别并提取最相关的信息。因此，系统只需对这部分规模更小且高度相关的文本进行嵌入和索引。这一策略使 KATS 能够构建内容丰富且目标明确的知识库，同时保持较小的存储占用。

总而言之，我们的效率与成本分析表明，KATS 是一个实用的系统。它能够有效平衡时间、应用程序接口调用和存储方面合理的一次性离线成本，从而提供高效且经济的在线查询功能。

## 7.4 增量更新效率

我们系统的一项关键要求是能够随着时间推移高效地纳入新知识。为评估增量更新处理流程的效率与可扩展性，我们在 CS-TDS$_M$ 语料库上开展了一项实验。首先，我们使用包含 518 篇文档的基础集合构建初始知识库。随后，我们通过分批添加其余文档来模拟新数据的到达，并测量每次更新操作所需的时间。

如图 6 所示，新文档数量与更新处理所需时间之间近乎呈线性关系。明显的趋势线表明，更新成本主要取决于传入数据的规模。这证实了我们的增量方法效率高且可扩展，并可避免随着语料库增长而执行完整系统重建所带来的高昂成本。

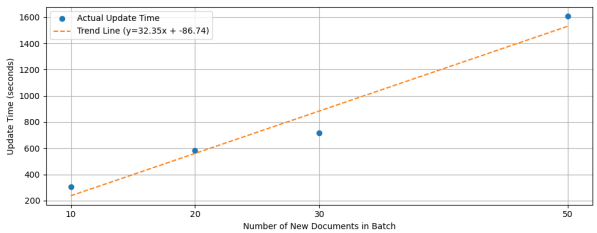



**图 6：新增文档数量与增量更新时间成本的关系，以及拟合得到的趋势线。**

## 7.5 消融研究

为了了解 KATS 系统中各关键组件的独立贡献，我们开展了一系列消融研究。在这些实验中，我们系统地禁用完整处理流程中的特定模块，并观察其对端到端有效性的影响。这样，我们便能量化系统各项设计选择所带来的价值。

*7.5.1* 大语言模型重排序。 我们首先研究最后一步基于大语言模型的重排序所产生的影响。图7中的蓝色和橙色折线

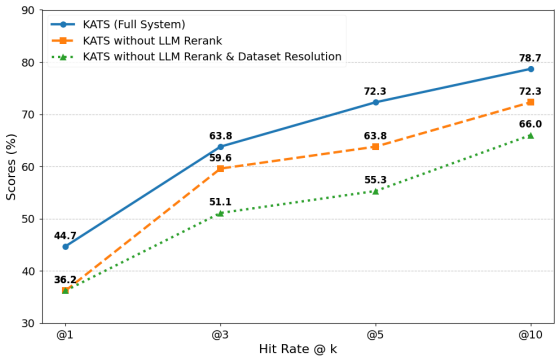



**图7：CS-TDS$_M$ 上消融实验的结果。**

比较了完整系统与禁用重排序模块的变体的性能。如图所示，加入大语言模型重排序器后，各级命中率均显著提升。这表明，尽管我们的基于图的排序能够有效生成一组优质候选项，但大语言模型的推理能力对于进一步优化排名靠前的排序结果的精确率至关重要。

*7.5.2* 数据集消歧。接下来，我们评估数据集消歧模块的贡献。该模块旨在解决实体歧义。为更好地展示该模块对知识图谱的影响，在实验中，我们不使用大语言模型重排序，而是直接评估来自知识图谱的搜索结果。如图7中的橙色和绿色折线所示，禁用该模块对排名第一的命中率影响甚微，但会导致其他命中率指标的性能显著下降。这一结果有力地验证了我们的假设：消解数据集命名歧义是现实世界数据集搜索中的关键因素。通过合并重复实体，我们的系统能够整合碎片化信息，构建更完整、准确的知识图谱，从而直接提升召回率和相关性。

## 7.6 微基准测试

除了端到端性能之外，我们还开展了一系列微基准测试，以分析 KATS 系统的鲁棒性和敏感性。具体而言，我们考察了系统在关键超参数方面的性能，以及其对底层大语言模型的依赖。

*7.6.1* 超参数敏感性。 我们首先分析系统对其关键超参数选择的敏感性，以确保其出色性能并非局限于某一狭窄设置。我们重点考察 PageRank 阻尼系数 ($\alpha$)，该参数用于平衡初始语义搜索与后续图遍历的影响。

为此，我们固定所有其他参数，并在 CS-TDS$_M$ 语料库上运行系统，同时改变 $\alpha$的取值：

$$\alpha \in \{0。75、0。80、0。85、0。90、0。95\}$$

记录其对命中率@3指标的影响。

如图8所示，在合理的 $\alpha$ 取值范围内，系统始终保持较高的有效性。较低的 alpha 值会更加侧重于图探索，从而导致性能略有下降。这表明，种子任务提供的初始语义信号




**图8：CS-TDS$_M$上，命中率@3对 PageRank 阻尼系数 ($\alpha$) 参数的敏感性分析。**

在实验中具有很高的价值。然而，随着超参数变化，整体性能仍相对稳定。这一结果表明，KATS 具有较强的鲁棒性，不会过度依赖对超参数 $\alpha$进行精确调优。

**表6：在 CS-TDS$_M$上，使用不同大语言模型进行构建和问答时 KATS 的命中率比较。**

| 构建 | QA | @1 | @3 | @5 | @10 |
|---|---|---|---|---|---|
| gpt-4o-mini | gpt-4o-mini | 0.447 | 0.638 | 0.723 | 0.787 |
| gemini-1.5-flash | gpt-4o-mini | 0.426 | 0.690 | 0.746 | 0.787 |
| gemini-1.5-flash | gemini-1.5-flash | 0.404 | 0.532 | 0.617 | 0.702 |

*7.6.2* 对大语言模型的鲁棒性。为验证我们的框架有效性并不依赖于某个特定的专有模型，我们使用另一种大语言模型评估其鲁棒性。具体而言，在构建和问答阶段，我们将 gpt-4o-mini [28] 替换为性能相当的模型 gemini-1.5-flash [13]，，并在 CS-TDS$_M$ 语料库上重新进行了端到端评估。

表6展示了这项敏感性分析的结果。研究结果有两点。首先，仅替换构建阶段使用的大语言模型时，KATS 的端到端性能仍然非常稳定，命中率指标没有显著下降。这有力地表明，我们的处理流程对抽取模型的具体选择并不敏感。

其次，这种稳定性表明，我们架构中的下游组件，即混合实体消歧和基于图的排序，足以应对不同抽取模型可能带来的变化和轻微错误。尽管在问答阶段使用较弱的模型会如预期般影响性能，系统仍优于基线方法。这项实验验证了 KATS 的核心优势在于其专门设计的架构和策略，而非依赖任何单一专有模型来完成知识抽取流程。

# 8 案例研究

## 8.1 增量知识发现

为展示增量更新功能的实际用途，我们通过一个案例研究来说明 KATS 如何动态地

纳入新知识，从而回答先前无法回答的查询。

在实验中，初始构建过程中故意未纳入 CS-TDS$_M$ 语料库中的一篇特定论文，该论文提到了 **CARLA 模拟器数据集** 。随后，发出了一个与该数据集相关的查询，如表 7 所示。由于缺少相关文档，系统无法找到并推荐 CARLA 模拟器数据集。

之后，通过增量更新流程将这篇被暂时搁置的论文加入语料库。处理流程正确且高效地将新提取的任务和数据集信息整合进现有知识库。再次发出完全相同的查询时，KATS 成功识别出 **CARLA 模拟器数据集**是该任务的相关结果。

**表 7：增量知识发现案例研究结果比较。**

| **查询：** “使用高分辨率视觉输入和专家动作训练城市驾驶场景模型。” | |
|---|---|
| **更新前** | 无法找到 CARLA 模拟器数据集。 |
| **更新后** | 将 CARLA 推荐为排名前二的结果。 |

这一真实世界案例研究表明，增量更新机制高效且有效。它使 KATS 能够通过整合新文档不断发展，从而在新研究不断涌现时，让其数据集推荐保持最新。

## 8.2 与在线研究智能体的比较

KATS 的一个相关比较对象，是新兴的在线人工智能研究代理范式，例如 Gemini 的 Deep Research 功能。虽然两种方法都旨在回答复杂查询，但它们代表了截然不同的技术路线。我们将通过以下案例研究来说明这些差异，如表8所示。值得注意的是，这种比较本质上有利于 Deep Research，因为包含真实答案对的论文对 KATS 隐藏，但基于网络的代理仍可访问该论文。

**表8：在线研究代理案例研究的性能比较。**

| Task | 方法 | 延迟 | 结果 |
|---|---|---|---|
| 案例1 | Gemini | 392s | 前6个答案中包含真实答案 |
| | KATS | < 20秒 | 真实答案排名第5 |
| 案例2 | Gemini | 400s | 真实答案位于第 6–8 位 |
| | KATS | < 20秒 | 真实答案排名第 3 |

**案例1：物理常识推理任务**。Gemini Deep Research 访问了 111 个网站，其中包括一些非定向且无关的网站，例如 Hugging Face Daily Papers。392 秒后，Gemini 给出了 6 个答案，其中包括真实答案“ Kodak数据集”。KATS 在 20 秒内给出了答案，真实答案排在前 5 位。

**案例2：图像压缩评估任务**。在 2 次尝试中，Gemini Deep Research 访问了约 100 个网站，并在约 400 秒内分别给出了 6 个和 8 个答案，其中包括真实答案“PiQA”。KATS 在 20 秒内给出了答案，真实答案排在前 3 位，而且这一结果在多次尝试中始终一致。

从上述两个案例可以看出，KATS 与在线研究代理遵循不同的范式。KATS 通过投入离线成本，将用户定义的特定领域语料库处理为持久知识库。这种方法在查询延迟、成本效益和可复现性方面具有显著优势。此外，其离线处理能力使其非常适合处理私有文档集合，而增量更新机制则使其能够高效整合新知识。相比之下，在线研究代理会针对每个查询动态浏览通用网络。这种方式能够提供实时信息，但会导致更高的延迟、更大的成本，且结果可复现性较低；这些结果来源于专业网站和大众兴趣网站的混合。

## 8.3 精炼的任务消歧

自动化知识抽取面临的一项重大挑战，来自涉及复杂研究论文的情况：这类论文会使用不同数据集，针对不同能力开展多项评估任务。简单的抽取方法可能会错误地将所有数据集关联到一个宽泛的任务，从而降低知识图谱的精确率。

为展示 KATS 应对这一挑战的鲁棒性，我们进行了案例研究。我们从 CS-TDS$_L$ 中选取了一篇论文，该论文在三个截然不同的领域训练并评估一个大语言模型：数学推理、自然语言理解和代码生成。对于这 3 项特定任务，该论文分别使用 GSM8K [9], GLUE [36], 和 HumanEval [8], 。

我们的多智能体提取框架成功消歧了这些上下文。KATS 并未提取“评估模型”这类通用任务，而是识别出具体任务并将其与相应的数据集关联起来，如表9所示。

**表9：任务消歧案例研究的结果。**

| 数据集 | 提取的任务描述 |
|---|---|
| GSM8K | *“[...] 多项自然语言理解任务”* |
| GLUE | *“[...] 多步数学文字题”* |
| HumanEval | *“[...]为编程任务生成正确的代码解答”* |

更重要的是，生成的知识图谱正确地将这些内容表示为三个彼此独立的任务-数据集分支，没有在它们之间建立错误的关联。因此，查询某个特定任务时，PPR算法不会给另外两个任务打高分，从而确保 KATS 返回的结果精确。该案例研究通过实证验证，KATS 的架构能够捕捉单个文档中的特定上下文关系，并证实了自动化知识图谱构建流程具有较高的精确率。

# 9 讨论

## 9.1 研究发现与启示

我们广泛的实验评估表明，KATS 是一个专为从非结构化文本中进行面向任务的数据集搜索而设计的系统，在有效性和效率方面均显著优于最先进的通用型检索增强生成框架。这一发现为数据集发现的未来发展带来了若干重要启示。

*9.1.1* 从关键词到意图：弥合语义鸿沟。 以往研究指出，一项主要挑战在于，用户面向任务的需求与传统系统要求的基于关键词的查询之间存在显著的“语义鸿沟” [17]。我们的研究展示了一种成功弥合这一鸿沟的新范式。KATS 接受开放式自然语言描述作为输入，将“查询构造”的认知负担从用户转移到系统，由系统解析复杂的任务描述并将其映射到相关数据集。未来的数据发现系统不应过于专注于优化关键词匹配，而应更多地致力于开发能够分析复杂用户意图的模型。

*9.1.2* 从通用型检索增强生成到专用型检索增强生成。 我们的结果还表明，尽管通用型检索增强生成架构提供了强大而灵活的框架，但针对特定问题设计专用系统仍能显著提升性能。对于面向任务的数据集搜索，这种专门化不仅涉及检索，还需要将实体歧义和任务到数据集的映射深度整合进核心处理流程。针对高价值垂直领域开发专用系统，是检索增强生成研究的一个可行方向。

## 9.2 局限性

尽管我们提出的 KATS 系统在面向任务的数据集搜索方面展现出显著的有效性，但我们也认识到仍存在一些有待未来解决的局限性。

*9.2.1* 知识来源的领域特异性。我们当前的处理流程，尤其是多智能体抽取框架，是针对科学文献的独特特征量身定制的。我们方法的一个关键前提是，这些文档通常会在同一处提及数据集，并详细描述其所应用的具体任务。然而，当应用于不符合这一前提的其他类型文档时，我们抽取策略的有效性可能会有所不同。尽管我们已扩展系统，使其能够整合不涉及具体任务的数据集描述（例如来自数据集仓库的描述），但对于其他类型的文档，抽取流程仍需进行重大调整。

*9.2.2* 对大语言模型保真度的依赖。我们承认，与许多当代自动化知识抽取系统一样[11, 20, 33], ，我们的知识图谱质量从根本上取决于上游大语言模型在抽取和实体消歧阶段的保真度。大语言模型可能产生错误，例如抽取不正确，这些错误会沿处理流程传播，导致图中出现错误的节点或边。这种依赖确实令人担忧。不过，KATS 的设计采用了特定策略来降低这些风险。首先，我们的多智能体抽取框架将复杂的端到端抽取任务分解为一系列更小、更易处理的子任务。通过为不同智能体分配明确的职责，我们降低了大语言模型在任何单一步骤中的认知负荷，从而尽可能减少幻觉或遗漏细节等错误。其次，在实体消歧等关键阶段，我们有意限制大语言模型的作用。我们没有让大语言模型承担对模糊实体进行聚类这一开放式任务，而是采用混合方法。算法方法首先生成高概率候选对，随后由大语言模型执行更简单、更可靠的二元验证任务。

表6所呈现的证据表明，替换构建阶段的大语言模型后，端到端性能依然稳定，这证明了这些缓解策略的有效性。尽管如此，当前系统仍缺少一个专门模块，无法自动验证所有大语言模型输出的真实性，也无法依据源文档纠正可能存在的错误，这仍是一项局限。

# 10 结论

面向任务的数据集搜索是数据专业人员一项至关重要但尚未得到满足的需求，面临着语义鸿沟长期存在、缺乏任务到数据集的映射与基准测试，以及实体歧义等挑战。为应对这些挑战，我们的工作做出了几项重要贡献。首先，我们开发了 CS-TDS 基准测试套件，填补了缺乏评估基准这一关键空白，从而能够对面向任务的数据集搜索系统进行严格评估。此外，我们提出了 KATS，这是一种新颖的端到端流水线，可在离线状态下自动构建高质量知识图谱，并配备专用的实体消歧机制，以解决缺乏任务到数据集的映射及实体歧义问题。尤为重要的是，我们为该流水线配备了高效的增量更新机制，将静态知识库转变为动态且可扩展的知识库。为应对长期存在的语义鸿沟这一挑战，KATS 配备了在线查询处理流水线，将向量搜索与基于图的排序相结合，根据此前构建的知识图谱生成高度相关且符合任务需求的数据集推荐。

我们相信，我们的工作为下一代数据集发现系统奠定了坚实基础，并提供了一个充满希望的蓝图。在此基础上，未来研究还有若干值得探索的方向。一个重要方向是将我们的框架从科学文献拓展到新的问题领域，例如从医学文本中关联症状与疾病。未来研究还可以着重提升技术能力，例如为知识图谱开发实时流式更新机制，并加入自动事实核查模块，以提高其可靠性。

## 致谢

ChenhaoMa部分获得了国家自然科学基金（项目编号：62302421）、广东省基础与应用基础研究基金（项目编号：2025A1515010439）以及香港中文大学（深圳）广东省大数据计算重点实验室的资助。Xiaolin han获得了国家自然科学基金（项目编号：62302397）、中国博士后科学基金会（项目编号：2025M774364）以及陕西省博士后科研项目（2025BSHSDZZ106）的资助。

# 参考文献


[1] Mubashara Akhtar、Omar Benjelloun、Costanza Conforti、Luca Foschini、Joan Giner-Miguelez、Pieter Gijsbers、Sujata Goswami、Nitisha Jain、Michalis Kara- mousadakis、Michael Kuchnik、Satyapriya Krishna、Sylvia Lesage、Quentin Lhoest、Pierre Marcenac、Manil Maskey、Peter Mattson、Luis Oala、Hamidah Oderinwale、Pierre Ruyssen、Tim Santos、Rajat Shinde、Elena Simperl、Arjun Suresh、Goeffry Thomas、Slava Tykhonov、Joaquin Vanschoren、Susheel Varma、Jos van der Velde、Steffen Vogler、Carole-Jean Wu和Luyao Zhang。2024。Crois- sant：适用于机器学习的数据集的元数据格式。arXiv预印本 arXiv:2403.19546 (2024)。https://arxiv.org/abs/2403.19546 发表于 NeurIPS 2024数据集与基准专题。[2] Qi An、Chihua Ying、Yuqing Zhu、Yihao Xu、Manwei Zhang和Jianmin Wang。2025。LEDD：大语言模型赋能的数据湖数据发现。arXiv预印本 arXiv:2502.15182 (2025)。https://arxiv.org/abs/2502.15182 [3] Muhammad Imam Luthfi Balaka、David Alexander、Qiming Wang、Yue Gong、Adila Krisnadhi和Raul Castro Fernandez。2025。Pneuma：借助大语言模型实现表格数据表示与检索的端到端系统。载于ACM SIGMOD数据管理国际会议论文集。https://doi.org/10.1145/3725337 arXiv:2504.09207 [cs.DB]。[4] Alex Bogatu、Alvaro A. A. Fernandes、Norman W. Paton和Nikolaos Kon- stantinou。2020。数据湖中的数据集发现。载于第36届IEEE国际数据工程会议（ICDE）论文集。美国达拉斯。https://doi.org/ 10 1109 48307 2020 00067 。/ICDE。。Dan Brickley、Matthew Burgess和Natasha Noy。2019。Google Dataset Search：在开放网络生态系统中构建数据集搜索引擎。载于2019年ACM Web Conference（WWW '19）论文集。ACM，美国纽约州纽约市，1365–1375。https://doi.org/10.1145/3308558.3313685[6] Sonia Castelo、Rémi Rampin、Aécio Santos、Aline Bessa、Fernando Chirigati和Juliana Freire。2021。Auctus：用于数据发现与增强的数据集搜索引擎。载于VLDB基金会会刊，第14卷。VLDB基金会， 2791 2794 10 14778 3476311 3476346 –。https://doi.org/./。Adriane P. Chapman、Elena Simperl、Laura Koesten、George Konstantinidis、Luis Daniel Ibáñez、Emilia Kacprzak和Paul Groth。2019。数据集搜索：一项综述。VLDB期刊29，1（2019），251–272。https://doi.org/10.1007/s00778-019-00564-x[8] Mark Chen、Jerry Tworek、Heewoo Jun、Qiming Yuan、Henrique Pinto、Jared Kaplan、Harri Edwards、Yuri Burda、Nicholas Joseph、Sandhini Agarwal等。2021。评估基于代码训练的大语言模型。arXiv预印本 arXiv:2107.03374 (2021)。[9] Karl Cobbe、Vineet Kosaraju、Mohammad Bavarian、Mark Chen、Heewoo Jun、Lukasz Kaiser、Matthias Plappert、Jerry Tworek、Jacob Hilton、Reiichiro Nakano、Christopher Hesse和John Schulman。2021。训练验证器解决数学文字题。arXiv预印本 arXiv:2110.14168 (2021)。[10] Matthijs Douze、Alexandr Krylov、Jeff Johnson和Hervé Jégou。2024。Faiss库。arXiv预印本 arXiv:2401.08281 (2024)。[11] Darren Edge、Ha Trinh、Newman Cheng、Joshua Bradley、Alex Chao、Apurva Mody、Steven Truitt、Dasha Metropolitansky、Robert Osazuwa Ness和Jonathan Larson。2024。从局部到全局：一种用于查询聚焦摘要的图检索增强生成方法。arXiv预印本 arXiv:2404.16130 (2024)。[12] Sainyam Galhotra、Yue Gong和Raul Castro Fernandez。2023。METAM：目标导向的数据发现。载于第39届IEEE国际数据工程会议（ICDE）。美国加利福尼亚州阿纳海姆。https://doi.org/10.48550/arXiv.2304.09068 arXiv:2304.09068 [cs.DB]。[13] Google。2025。Gemini 1.5 Flash [多模态大语言模型]。https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/1-5-flash。[14] Bernal Jiménez Gutiérrez、Yiheng Shu、Yu Gu、Michihiro Yasunaga和Yu Su。2024。HippoRAG：神经生物学启发的大语言模型长期记忆。载于神经信息处理系统进展（NeurIPS）第37卷。https://arxiv.org/abs/2405.14831 NeurIPS 2024。[15] Xuming Hu、Shen Wang、Xiao Qin、Chuan Lei、Zhengyuan Shen、Asterios Katsifodimos、Christos Faloutsos、George Karypis、Lijie Wen和Philip S. Yu。2023。基于表格表示学习的自动表格并集搜索。（2023）。
h
t
t
p
s://
w
w
w.
a
mazon.science/publications/auto
matic-table-union-search-with-tabular-representation-learning [16] Zezhou Huang、Jiaxiang Liu、Haonan Wang和Eugene Wu。2024。快速且私密：基于任务的数据集搜索。载于创新数据系统研究会议（CIDR）。https://doi.org/ 10 48550 2308 05637 2308 05637 。/ [ ] arXiv。。arXiv：。cs.DB。[17] Madelon Hulsebos、Wenjing Lin、Shreya Shankar和Aditya G. Parameswaran。2024。“这比我预想的花了更长时间”：为什么数据集搜索仍然如此困难？载于2024年人在回路数据分析研讨会（HILDA ' 24 1 4 10 1145 3665939 3665959 ）论文集。ACM，智利圣地亚哥 [18] 2022 ，–。https://doi.org/./。Daniel Jarrett、Alihan Hüyük和Mihaela van der Schaar。。在线决策中介。载于神经信息处理系统进展（NeurIPS）。[19] Glen Jeh和Jennifer Widom。2003。扩展个性化网络搜索。载于第12届万维网国际会议（匈牙利布达佩斯）（WWW '03）论文集。A ssociation for C omput ing Machi nery, New York, NY, USA, 271–279. https://doi.org/ 10 1145 775152 775191 . / . Bernal Jiménez Gutiérrez、Yiheng Shu、Weijian Qi、Sizhe Zhou 和 Yu Su。2025。《从检索增强生成到记忆：大语言模型的非参数持续学习》。*arXiv* 预印本 *arXiv:2502.14802* (2025)。[20] [21] Vladimir Karpukhin、Barlas Oguz、Sewon Min、Patrick Lewis、Ledell Wu、Sergey Edunov、Danqi Chen 和 Wen tau Yih。2020。《面向开放领域问答的稠密段落检索》。载于*2020*年自然语言处理实证方法会议（*EMNLP*）论文集。计算语言学协会，线上，6769–6781。https://doi.org/10.18653/v1/2020.emnlp-main. 550 [22] Alex Krizhevsky。2009。从微小图像中学习多层特征。技术报告。[23] Patrick Lewis、Ethan Perez、Aleksandra Piktus、Fabio Petroni、Vladimir Karpukhin、Naman Goyal、Heinrich Küttler、Mike Lewis、Wen-tau Yih、Tim Rocktäschel、Sebastian Riedel 和 Douwe Kiela。2020。《面向知识密集型自然语言处理任务的检索增强生成》。载于神经信息处理系统进展第*33*卷（*NeurIPS 2020*）。https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html[24] Hao Liu、Zhengren Wang、Xi Chen、Zhiyu Li、Feiyu Xiong、Qinhan Yu 和 Wentao Zhang。2025。HopRAG：面向逻辑感知检索增强生成的多跳推理。arXiv:2502.12442 [cs.IR] https://arxiv.org/abs/2502. 12442 arXiv 预印本 arXiv:2502.12442，版本 2，最后修订于 2025年5月26日。[25] LlamaIndex 团队。2025。LlamaParse。https://github.com/run-llama/llama_cloud_ services。[26] Fatemeh Nargesian、Erkang Zhu、Ken Q. Pu 和 Renée J. Miller。2018。开放数据上的表联合搜索。*VLDB*基金会会刊11, 7 (2018), 813 825 10 14778 3192965 3192973 – 。 [27] https://doi.org/ . / . Sebastian Neumaier、Jürgen Umbrich 和 Axel Polleres。2016。开放数据门户中元数据的自动化质量评估。*ACM*数据与信息质量期刊（*JDIQ*） 8, 1 2016 2 10 1145 2964909 (11月)，。[28] 2024 4 [ ] https://doi.org/ . / OpenAI。. GPT- o mini 大语言模型。https://platform.openai.com/docs/models/gpt-4o-mini。[29] OpenAI。2024。text-embedding-3-small。https://platform.openai.com/docs/ guides/embeddings/embedding-models。[30] Norman W. Paton、Jiaoyan Chen 和 Zhenyu Wu。2023。《数据集发现与探索：综述》。计算机调查( 2023 10 1145 3626521 )。[31] [ ] https://doi.org/ . / pypdf 开发者。无日期。 . pypdf。https://github.com/py-pdf/pypdf。[32] Deepak Rajan、Michael J. Franklin、Stephan Hacker、Joseph M. Hellerstein 和 Joseph E. Gonzalez。2018。Aurum：一个数据发现系统。载于*2018*年国际数据管理会议（*SIGMOD*）论文集。ACM，2109–2112。https://doi.org/10.1145/3183713.3196927[33] Parth Sarthi、Salman Abdullah、Aditi Tuli、Shubh Khanna、Anna Goldie 和 Christopher D. Manning。2024。RAPTOR：面向树组织检索的递归抽象处理。载于国际学习表征会议（*ICLR*）。[34] Schema.org 社区。2011。Schema.org。https://schema.org。[35] Nandan Thakur、Nils Reimers、Andreas Rücklé、Abhishek Srivastava 和 Iryna Gurevych。2021。BEIR：用于信息检索模型零样本评估的异质基准。载于第三十五届神经信息处理系统大会数据集与基准赛道（第二轮）。https://arxiv.org/abs/2104.08663[36] Alex Wang、Amanpreet Singh、Julian Michael、Felix Hill、Omer Levy 和 Samuel R. Bowman。2019。GLUE：自然语言理解的多任务基准与分析平台。载于 ICLR 论文集。[37] Zixin Wei、Jun Han、Xiaolin Han 和 Chenhao Ma。2025。SemExplorer：一种用于语义化定制数据集搜索的用户界面。载于*2025*年国际数据管理会议伴随会议（德国柏林）*(SIGMOD/PODS '25)*。计算机协会，美国纽约州纽约市， 255 258 10 1145 3722212 3725133 – 。https://doi.org/ . / . [38] Zhilin Yang、Peng Qi、Saizheng Zhang、Yoshua Bengio、William W. Cohen、Ruslan Salakhutdinov 和 Christopher D. Manning。2018。HotpotQA：用于多样且可解释的多跳问答的数据集。载于*2018*年自然语言处理实证方法会议论文集。计算语言学协会，比利时布鲁塞尔，2369–2380。https://doi.org/10. 18653/v1/D18-1259[39] Wayne Xin Zhao、Kun Zhou、Junyi Li、Tianyi Tang、Xiaolei Wang、Yupeng Hou、Yingqian Min、Beichen Zhang、Junjie Zhang、Zican Dong、Yifan Du、Chen Yang、Yushuo Chen、Zhipeng Chen、Jinhao Jiang、Ruiyang Ren、Yifan Li、Xinyu Tang、Zikang Liu、Peiyu Liu、Jian-Yun Nie 和 Ji-Rong Wen。2023。《大语言模型综述》。https://arxiv.org/abs/2303.18223。[40] Yingli Zhou、Yaodong Su、Youran Sun、Shu Wang、Taotao Wang、Runyuan He、Yongwei Zhang、Sicong Liang、Xilin Liu、Yuchi Ma 等。2025。《统一框架下对基于图的检索增强生成的深入分析》。*arXiv* 预印本 *arXiv:2503.04338* (2025)。