ModelTables：关于模型的表格语料库 s

Zhengyuan Dong 滑铁卢大学
加拿大安大略省滑铁卢
zhengyuan.dong@uwaterloo.ca

Victor Zhong 滑铁卢大学
加拿大安大略省滑铁卢
victor.zhong@uwaterloo.ca

Renée J. Miller 滑铁卢
大学 加拿大安大略省滑铁卢
rjmiller@uwaterloo.ca

arXiv:2512.16106v1 [cs.DB] 18 Dec 2025

摘要

我们介绍*ModelTables*，这是一个针对模型湖中表格的基准，旨在捕捉文本检索往往忽略的性能表和配置表的结构语义。该语料库由 Hugging Face 模型卡、GitHub README 文件和引用的论文构建而成，并将每个表格与其周边的模型及出版物语境相关联。与开放数据湖表相比，模型表规模较小，但呈现出更密集的表间关系，反映了模型与基准的紧密协同演进。目前发布版本涵盖超过 6 万个模型和 9 万个表格。为评估模型与表格相关性，我们利用三种互补信号构建多源真实标注：（1）论文引用链接；（2）明确的模型卡链接和继承关系；（3）共享训练数据集。我们展示了该基准的一个全面实证用例：表搜索。我们在此基准上比较了数据湖的经典搜索算子（可并集、可连接、关键词）和信息检索基线（稠密、稀疏、混合检索）。基于并集的语义表检索总体 P@1 达到 54.8%（引用信号为 54.6%，继承关系信号为 31.3%，共享数据集信号为 30.6%）；基于表格的稠密检索达到 66.5% 的 P@1，元数据混合检索达到 54.1%。该评估表明，改进表搜索方法仍有显著空间。通过发布 ModelTables 及其构建流程，我们提供了首个描述人工智能模型的结构化数据大规模基准。我们在模型湖中的表格发现用例，为开发更准确的语义检索、结构化比较以及对结构化模型知识进行有原则的组织提供了直观认识和实证依据。源代码、数据及其他资源已发布于https://github.com/RJMillerLab/ModelTables。

1 引言

许多表格语料库由研究人员创建并共享。这些语料库包括 Webtables [6, 14, 30, 32, 47], WikiTables [3],VizNet [24], GitTables [25], 以及若干开放数据语料库 [43, 62]其中包含从开放数据门户抓取的表格。这些表格集合已被用于研究各种数据管理任务，包括语义类型标注（确定属性的语义类型）、模式补全、表标注、模式匹配、实体匹配和表搜索（查找相关表格）。GitTables 等大型语料库旨在用于机器学习中的表解释或理解任务 [25]。

相比之下，在这项工作中，我们介绍了 ModelTables，这是首个描述人工智能模型的表格语料库。不同于以往主题覆盖面广、内容丰富的表格语料库，我们专注于描述模型的表格。为了创建这一语料库，我们利用了三类资源。首先是模型仓库（或模型湖 [45]），其中包含大量由模型卡 [41,54] 描述的模型，以及描述模型开发中使用的训练数据和基准的数据卡 [20, 33]。其次是包含模型相关代码的代码仓库。模型卡通常会链接到代码仓库。第三是论文仓库（更具体地说，对于我们的基准而言，是两个论文仓库），其中包含描述这些模型的学术论文。模型卡和数据卡以及代码仓库都可以链接到这些论文。请注意，在我们的语料库中，每个表格都与一个模型相关联（如果存在重复项，一个表格也可能与多个模型相关联）。

1.1 从语料库到基准

请注意，所提到的大多数语料库本身并不是基准，因为它们通常不包含针对所研究任务的真实标注。相反，研究人员通常会从这些语料库中人工整理子集，或手动处理其中的表格，以创建带有真实标注的标注基准。

例如，经过广泛研究的相关表搜索。相关表格通常被定义为“包含可能原本可以放在同一张表格中的内容”的表格 [51],，基于这一理念，大多数相关表搜索工作都专注于查找可以进行并集或连接操作的表格 [4, 12, 43, 61]。为了构建基准，会创建一组带标签的表（其中包含已知相关和不相关的表格），例如表并集搜索基准 TUS [43] 和 SANTOS [27]，其标签由人工整理；以及 UGEN [46]，其标签由大语言模型生成。

相比之下，我们发布了 ModelTables，其中包含若干种用于表格相关性的基线真实标注，这些标注由模型开发者自己生成。与其他语料库不同，我们提议使用 ModelTables 研究表格相关性的主题性概念，而非结构性概念 [51]。

我们的 相关性真实标注来自三个来源。首先，模型卡彼此链接。第一种相关性度量是模型卡相关性：某个模型与其模型卡所链接的任何模型相关（通过 URL 或 base_model 这样的标签）。其次，数据卡也会彼此链接，并链接到其他模型。第三，论文相关性通过论文中的引文得到了充分研究，而模型卡也会链接到论文。

1.2 基准详情

我们基于 Hugging Face Model Lake 构建了一个大规模基准[1]，并从模型卡或数据卡中提取所有表格。如今，Hugging Face 是最大的开放模型仓库 [20, 33, 41, 54]。模型卡（数据卡）通常会引用 GitHub 网站，我们将 GitHub 用作代码仓库。我们还从 GitHub README 文件（模型卡或数据卡引用的文件）中提取表格。最后，我们从模型卡（或数据卡）或相应 GitHub 网站引用的学术论文中提取表格。在我们的工作中，我们

[1]huggingface.co

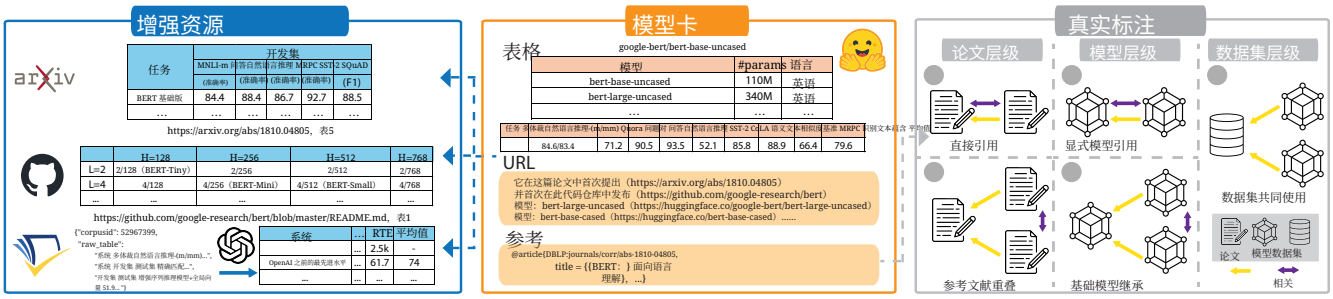


图1：模型湖基准设置流程。本图概述了我们用于建立模型湖基准的端到端自动化工作流程。图中展示了收集、清理和关联多种表格的通用流程，包括整合论文层级引文图。此外，图中还详细说明了我们如何结合论文引文、模型卡中的模型谱系以及共享数据集元数据中的信号，构建用于模型关联性的多层级真实标注，为评估提供稳健基础。

发现了许多指向 arXiv 论文的链接[2]，以及 BibTeX 条目和简单的论文标题（通常见于表格图注中）。对于未发布在 arXiv 上的论文，我们尝试在 Semantic Scholar 中查找论文源文件。我们使用官方 Semantic Scholar 语料库[3]，这是一个每周更新、可通过 Semantic Scholar API 访问的资源。其元数据和引用提取遵循 S2ORC 流水线 [38]，，并在当前部署版本中进行了商用级别的改进。

1.3 贡献

本研究做出了多项贡献。（1）我们提出了首个用于描述人工智能模型的表格大规模基准。不同于旨在涵盖广泛主题内容的既有表格基准，我们的表格全部用于描述模型，因此可用于模型理解任务。（2）我们引入了一组相关性真实标注，这些标注基于学术论文、模型卡和数据集中可验证的相关性信号，使我们的基准能够用于各种相关性搜索任务。（3）我们介绍了一套可复现的基准创建流水线。尽管我们的基准使用开放模型、代码和论文，这套流水线也可在内部使用，以便为私有模型湖创建资源。在这种情况下，论文相关性将替换为项目或团队相关性。[4] 迄今为止，鲜有研究帮助数据驱动型组织管理和理解自身丰富的私有模型湖。我们的工作迈出了第一步，展示了如何利用模型湖中的结构化数据来理解模型。（4）我们分析了该基准及相关性图，并将其与最先进的表格搜索语料库和基准进行比较。我们的基准包含更多相关表格，其相关性分布也更丰富，能够反映模型的真实特性（有些模型是枢纽，与许多模型相关；相关模型数量的分布呈偏态，许多模型仅与少数模型相关）。（5）我们通过实证展示了现有相关表格搜索方法（来自信息检索和数据湖领域）在模型表上的性能。结果表明，在查找主题相关表格方面仍存在研究空白。我们展示了如何利用该基准研究新的表格相关性形式（超越典型的结构或信息检索概念）。（6）我们讨论了该基准在模型理解和表格理解方面的若干用例。结果表明，语义搜索和表格发现从互补的视角揭示表格相关性，从而能够更深入地理解模型相似性。

论文组织。 在第 3 节中，我们介绍基准构建——如何查找和提取表格。在第4节中，我们介绍真实标注提取。我们定义了三个层级的真实标注，分别由模型开发者提供的不同信号构建而成。在第5节中，我们分析该基准，提供有关表格的统计信息，分析不同的相关性图，并将其与其他表格基准（包含真实标注）进行比较。在第6节中，我们将相关表搜索作为该基准的一项应用。我们表明，现有的信息检索/数据库表格搜索方法无法很好地捕捉表格相关性，并指出了新的研究方向。在第7节中，我们讨论该基准的其他应用。

2 相关工作

2.1 数据发现

在数据湖中检索结构化表格通常可表述 [7, 59]为：给定一个查询表，系统必须根据模式兼容性、内容重叠或语义相似性，从大型异构存储库中发现相关表格。早期工作聚焦于可连接表格发现，其中 DeepJoin [12] 使用预训练语言模型嵌入列值，并通过最近邻搜索执行句法和语义连接；而 LSHEnsemble [62] 和 Josie [61] 则采用重叠集合索引，高效识别共享连接键的表格。Nargesian 等人提出了可并集表搜索方法 [43]。SANTOS 在此基础上进行了扩展 [27]，利用知识库驱动的语义关系来改进并集预测；D3L 则使用正则表达式 [4]。与此同时，基于关键词的表搜索方法会为列名和元数据建立索引，以支持在数据湖中进行临时文本查询。Blend [15] 统一并优化了这些经典的数据湖任务。此外，表格相关性技术通过分析属性共现统计量，量化表格之间的语义和结构关联，从而提高相关表格检索的精确率 [6]。

近期研究引入了超越简单检索的任务。表格回收 [16] 旨在找回查询表中缺失的行

[2]arxiv.org [3]https://www.semanticscholar.org/[4]LinkedIn Engineering Blog — LinkedIn 的机器学习生产力扩展；Netflix TechBlog — 支持 Netflix 的多样化机器学习系统。

或列：通过有选择地合并相关表格中的内容，有效地“修复”不完整数据集。列消歧 [2] 通过将不一致或含义模糊的表头名称关联到规范类型或实体，从而改善不同来源之间的互操作性。表格增强 [57] 则从外部存储库中提取互补属性（如地理或人口统计数据），以丰富现有表格。通过为经典的连接、并集和关键词搜索扩展这些新能力，数据湖在下游分析和人工智能工作流方面变得更加强大、灵活。

2.2 模型湖

模型湖仍处于早期发展阶段 [19, 45]。Pal 等人[44] 提出了模型湖的愿景，其中包括四项核心任务：归因，即将模型的输出追溯到有影响力的训练数据或内部特征；版本管理，即重建有向模型树，以揭示父子派生关系；搜索，即根据模型的固有属性、输出行为或元数据定位模型；以及基准测试，即使用统一测试套件大规模评估模型。Garouani 等人 [19] 提出了三区架构（摄取、分析、治理），并为数据和模型构建了丰富的元数据模型。

Hugging Face 可能是目前规模最大的开放模型湖。Hugging Face 支持按模型名称进行关键词搜索，也支持对文本元数据进行全文搜索。最近，他们新增了语义搜索功能，该功能使用大语言模型生成的数据集和模型摘要，并支持对这些摘要进行相似度搜索。[5] 这是一种黑箱搜索，因此无法纳入我们的比较。我们的工作对此进行了补充，提供了一个开放基准，用于评估其搜索以及其他语义模型和数据集搜索方法。

2.3 科学表格

在其他领域，表格结构化数据也催生了专门的基准和“湖”概念。在数据库社区，GitTables [26] 旨在支持列类型检测和模式推断，提供了一个从 GitHub 提取的百万表格语料库，以及一个较小的带标签子集，用于评估语义标注模型。在自然语言处理领域，各种表格问答基准不断涌现。SciTab [39]通过将经专家验证的科学陈述与其源表格配对，专注于事实核验；TabLeX [10] 则评估从表格图像中提取结构和恢复内容的能力，提供了广泛的推理挑战。与此同时，在科学研究领域，SciSciNet 等开源数据湖 [35] 以及欧洲核子研究中心的 ESCAPE 将引文相关元数据和实验结果汇集到统一存储库中；尽管采用了“湖”式架构，其任务主要聚焦于大规模学术分析和数据编排，而非细粒度的表格理解或模型选择。更具体地说，科学数据湖中的表格包含多种功能类型，例如性能表、配置表和示例表格；与之相应的任务包括语义分类 [29, 31], 结构分类 [22], 以及列级标注 [25] 或分类 [31, 55]。

[5] https://huggingface.co/spaces/librarian-bots/huggingface-semantic-search

2.4 引文图

引文图分析是一个研究充分的领域，涉及引用意图分类、链接预测、推荐和科学计量影响评估等任务。Crossref [23], OpenAlex [48] 和 Semantic Scholar [18] 等众多平台提供功能强大的 API 和原始数据转储，涵盖出版物、资助项目、专利、临床试验及媒体中的数十亿条引文链接。科学学数据湖 SciSciNet [35]在这些基础上，将多种来源整合到统一的分析框架中，以研究研究影响力、合作模式和科学演进。早期工作主要关注引文关系分类，例如 SciCite [9] 将引文语境标注为背景、方法论、结果或其他类别；也关注利用图嵌入和神经网络模型进行引文链接预测。推荐系统利用这些链接结构来呈现相关文献，而基于引文的指标则量化影响模式并揭示新兴趋势。与此同时，数据引文研究通过数据引用 [5] 将数据集视为图中的一等节点，并通过与版本管理任务无缝集成，丰富数据湖和模型湖中的模型选择信号。通过统一出版物和数据集引文，这些方法能够带来跨领域洞见，连接学术交流、数据共享与可复现性。

3 基准构建

本节详细介绍了我们基于模型湖构建模型表格基准的过程。我们首先界定支撑最终基准可靠性和有效性的基础假设与范围。在此框架下，实际构建分为几个关键阶段：全面的元数据提取、严格的表格质量控制，以及利用模型卡、代码仓库和学术论文等多种来源的信号，为表格相关性建立真实标注。我们介绍了多阶段表格爬取流程（图1），旨在系统地收集表格及相关元数据，包括引文和 URL，从而建立模型与数据集之间丰富的关系。

3.1 表格提取

我们通过收集模型卡 README 中包含的表格，以及模型卡所引用的其他资源中的表格，构建基准。我们使用 HuggingFace 模型卡数据集[6] 作为主要来源。Huggingface 是一个大型且广受欢迎的模型湖（在我们抓取时拥有超过 100 万个模型），提供模型制品的集中托管服务和丰富的元数据。我们从每张模型卡中提取三个核心元素：嵌入式表格、URL 和 BibTeX 参考文献。随后，对于每个 URL 和 BibTeX 条目，我们识别其来源平台（arXiv、GitHub 或其他 Hugging Face 模型卡），并应用相应的表格抓取策略。最终的基准包含来自四种资源的表格。该提取流程在算法 1 中进行了形式化描述，具体内容如下。

模型卡表格。 我们首先解析模型卡中直接嵌入的所有表格。这些表格很容易解析，因此

[6] https://huggingface.co/datasets/librarian-bots/model_卡片_含_元数据250118

算法 1： 从模型库构建表格语料库

输入： 来自 Hugging Face 的全部模型卡集合 M C_{all}
输出： 经过整理的表格集合 T，并链接至其来源 M C

```
T ← ∅
MC_valid ← 筛选 M C_all，保留包含论文和表格的模型卡
对每个 mc ∈ MC_valid 执行
    T_card ←提取表格(mc.readme)
    将 T_card 中的全部表格添加到 T，并将它们链接到 mc
    P_refs ← 提取论文参考文献(mc.readme)
    对每个 p ∈ P_refs 执行
        T_arxiv ← 提取表格(GetArxivContent(p) )
        将 T_arxiv 中的全部表格添加到 T，并将它们链接到 mc
        T_s2orc_raw ←获取 S2ORC 文本表格(p)
        T_s2orc_proc ← 恢复表格结构(T_s2orc_raw)
        将 T_s2orc_proc 中的全部表格添加到 T，并将它们链接到 mc
    G_urls ←提取 GitHub URL(mc)
    对于每个 g ∈ G_urls 执行
        T_github ← 提取表格(获取 README(g) )
        将 T_github 中的全部表格添加到 T，并将它们链接到 mc
返回T
```

非常规整。表格通常是配置表或性能表，但我们纳入模型卡中的所有表格。

GitHub 表格。 如果模型卡链接到 GitHub 仓库，我们会下载 README 文件并提取其中找到的所有表格。与模型卡中的表格一样，这些表格很容易提取，而且相当干净。一些 GitHub URL 指向大型检查点或缺少 README 文件、也没有可靠位置可查找结构化表格的代码仓库，因此不会纳入我们的基准。

arXiv 表格。 如果模型卡链接到一篇或多篇 arXiv 论文（或包含某篇 arXiv 论文的 BibTex 文件），我们希望使用这些论文中的表格。如果 arXiv 链接提供了适合可靠提取表格的 HTML 版本，我们就从标记中提取并使用这些表格。目前，我们发现直接从 PDF 文件中提取表格无法得到高质量的结果，因此我们的基准不包含这些表格。为避免纳入无关论文和表格，我们明确排除占位论文链接[7],该链接出现在 Hugging Face 的模板模型卡中，许多人会以此为起点，却忘记删除[8]。

Semantic Scholar 表格。 如果模型卡链接到 Semantic Scholar 上的一篇论文[9], 我们便利用 Semantic Scholar 开放研究语料库（S2ORC） [38]。S2ORC 通过其处理流程处理 PDF 和 LaTeX 文件，并将识别出的表格、图表和文本部分保存到其公开的数据转储中。然而，这些提取出的表格只是缺乏行列结构的非结构化原始文本。因此，从 S2ORC 数据中获取这些原始表格文本后，我们会将其输入大语言模型（LLM）进行解析和完善。LLM 的任务是将这些非结构化文本块转换为整洁、结构统一的表格格式。

[7]https://arxiv.org/abs/1910 09700 8.
https://huggingface.co/templates/model-card-example [9]
https://www.semanticscholar.org

3.2 抽取质量控制

我们最初从多个平台收集表格，但这些表格在质量和一致性方面面临着显著挑战。为了创建高质量且稳健的基准，我们的后处理流程通过三个关键阶段完善数据。首先，我们进行**质量控制**，以纠正底层解析模式。随后进行**策略性筛选**，移除不适合分析的整张表格。最后，经过完善的数据集会接受**系统性数据增强**，以应对表示形式的异质性。

表格的质量控制。 来自不同平台的表格质量差异显著。这些表格具有显著的结构异质性。它们源自 HTML 和 Markdown 等格式，其底层结构类似于常见的网页表格和 CSV 文件，并且可能具有复杂的布局，例如多级表头或合并单元格。我们的抽取流程依赖广泛使用的最先进表格抽取方法 [8, 21]，这些方法专为处理高度异质的表格而设计。

具体而言，我们的流程能够处理若干棘手的边缘情况。它解决了 列对齐问题，包括由合并或空单元格导致的问题。它通过专门解析包含竖线字符（'|'）的 Markdown 单元格来处理内部定界符冲突，这种情况存在于一些 Hugging Face 模型卡中。 [10] 该流程通过将脚注文本直接合并到相应单元格中，保留 单元格级注释 中的上下文。它还会拼接分散的 多页表格，并清除诸如空行之类的视觉格式伪影。最后，对于特殊字符和符号（例如，`\lambda`，`\cdot`），我们会原样保留。鉴于领域专用记号数量庞大，保留原始表示形式比尝试进行全面规范化更能稳健地维护语义完整性。这些有针对性的修正确保所有表格在结构上都完整无误。

策略性筛选。 在纠正解析错误后，我们采用策略性筛选流程，移除存在严重结构问题或不适合按列分析的表格。具体而言，基于大语言模型的恢复过程从 Semantic Scholar 提取结果中得到的表格质量参差不齐，因此这些表格仅适用于消融研究，而不适合作为我们的主程序结果。

3.3 表格增强

模型湖面临的一项关键挑战是，不同来源中表格的表示方式存在显著异质性。科学表格在方向、格式和内容编码方面差异很大，这阻碍了直接的匹配与检索。为解决这一问题，我们采用了两种旨在提升表检索准确率的增强技术。根据具体使用场景，通过这些增强技术创建的表格可以选择性地添加到基准中。

转置增强。 在模型湖中，或者更广泛地说，在科学表格中，作者可能会横向呈现信息。例如，表格中的每一列对应一组基准数据集中的一个数据集，列值则表示某个配置在该数据集上的性能。相反，在比较单个数据集的不同指标时，常采用纵向布局，并将指标排列为行标签。这些风格上的差异意味着

[10]https://huggingface.co/spacy/nb_core_news_lg

RoBERTa - 表10

超参数	RACE	SQuAD	GLUE
学习率	1.00E-05	1.50E-05	{1e-5, 2e-5, 3e-5}
批次大小	16	48	{16, 32}
权重衰减	0.1	0.01	0.1
最大训练轮数	4	2	10
学习率衰减	线性	线性	线性
预热比例	0.06	0.06	0.06

DPS - 表2

模型	数据集	批次大小	学习率	训练轮数/步数	预热比例/步数	分层学习率衰减	
BERT	all	16	2.00E-05	3 个训练轮次		10%	-
RoBERTa	RTE	16	2.00E-05	2036 步		122 步	-
	MRPC	16	1.00E-05	2296 步		137 步	-
	语义文本相似度基准	16	2.00E-05	3598 步		214 步	-
	CoLA	16	1.00E-05	5336 步		320 步	-
…	…	…	…	…	…	…	…

图2：来自 RoBERTa 的表格 [37] 和 DPS [58] 启发我们采用增强方法：通过转置实现对齐，并通过表头—单元格融合实现语义规范化

表示相似信息的表格，其表头和关键标签的方向可能不同。例如，如图 2所示，在一张表格中作为列标题的内容，在另一张表格中可能是行标题。对于来自四个来源的表格，我们分别创建一组表格，其中包含每张表格的转置版本。

表头到单元格增强。此外，我们观察到，有些表格会将表头语义直接嵌入单元格值中。一种典型情况是，某些列将类别标签与数值数据结合起来——例如，“训练轮数/步数”列可能包含“3 个轮次”或“1500 步”这样的条目，在一个单元格中融合数量与单位。这种做法虽然提高了人类的可读性，却给自动解析和语义对齐带来了挑战，因为同一概念属性在不同表格中可能被拆分到不同的结构元素中。我们可以通过将数值与表头名称拼接，为解读表格中的数值提供更多语义上下文，从而缓解这一问题。给定来自四种来源中任意一种的表格集合，我们会对该集合进行表头增强：对于每个表格，将表中的每个值与其属性表头名称拼接。

4 模型与表格相关性

除了提取表格，我们还提取模型 ID 和论文 ID。因此，我们的语料库包含一个异构图，其中表格、模型和论文是节点。所有表格都至少与一个模型相关联（概念上通过一条边连接；如果存在重复表格，则可能关联多个模型）。一个模型可能与一篇或多篇论文相关联（通过其模型卡、相关数据卡或相关代码仓库）。每篇论文都至少与一个模型相关联。这些直接的“提取”链接或边为关联表格和模型提供了一种方式。在本节中，我们将探讨更多语义层面的关联性。具体而言，我们希望利用这个异构图中的直接关联以及可能存在的间接关联，构建多个相关性图，以用于表格和模型搜索及理解任务。

我们的核心假设是，ModelTables 可用于各种表格和模型理解任务。例如，通过查找并整合与某个模型相关的表格，我们可以更好地回答复杂的数据科学问题，例如“最适合我的模型是什么任务？”因此，我们面临的一个关键挑战是在基准中找到相关表格和相关模型。这引出了一个根本问题：我们如何判定两个模型确实相关？为此，我们构建了几种可能的模型相关性图。指导这些图构建的核心概念在表 1中定义。

表1：模型与表格相关性的定义。

概念、定义与检测
模型关联性： 基于模型卡或其训练数据形成的模型间关联。它通过以下位置的信号建模：论文层级（引文、共同参考文献）、模型层级（直接模型卡链接或由模型标签表明的谱系）以及数据集层级（共同数据集）。
表格相关性： 表格之间推断出的关系。它根据提取位置进行建模，并继承与之关联的源模型或论文的相关性状态与每个表格。

我们从三个关键层面评估模型关联性，如图 1 所示：论文、模型和数据集。

4.1 论文相关性

首先，在论文层级，我们从 Semantic Scholar 数据中获取每篇论文的参考文献和引文。根据这些信息，我们判断两篇论文是否直接引用对方，或是否存在重叠的参考文献列表，以确定它们的相关性。

更正式地说，如果模型 m_i 和 m_j 的任意相关论文通过以下方式相关，则这两个模型具有论文相关性（R_{paper}）：

（a）直接：直接引用 ——一篇论文直接引用另一篇；

（b）重叠：参考文献重叠 ——它们的参考文献列表中至少有一篇共同引用的文献。参考文献重叠是通过比较参考文献列表计算得出的，以捕捉直接和潜在的引用关系。例如，几乎同时发表的两篇论文可能高度相关，但彼此之间未必相互引用。不过，参考文献重叠可以揭示这种关系。

引文元数据通过 Semantic Scholar API 检索[11]，包含两个关键属性：引用意图和*isInfluen- tial* [9, 52]。这些意图引文来自论文的方法论或结果章节。与背景或相关工作章节中的引文相比，这些引文被认为更有意义（例如，后者的分类依据是一个多类分类模型，该模型会对每条引文的修辞角色进行分类）。*isInfluential* 是一个二元指标，用于检测引文是否似乎具有重要性。该指标由一个二元分类模型生成，该模型综合了多个文本层面和引文层面的因素，例如引文频率、章节位置和上下文强调 [9, 52]。

在各种引用关系中，直接引文在某种意义上比重叠引文更严格，但这两种关系会在模型（或其表格）之间形成不同的相关性图。在每组（直接引文与重叠引文）中，我们

[11] https://www.semanticscholar.org/product/api/tutorial

可以将考察范围限定为同时具有有影响力和意图属性的引文、具有其中一种属性的引文，或所有引文。因此，我们的模型图中的论文相关边可根据引文类型构成八种可能的相关性图：直接（意图 ∩ 有影响力）；直接（意图）；直接（有影响力）；直接；重叠（意图 ∩ 有影响力）；重叠（意图）；重叠（有影响力）；重叠。

使用论文相关性（R_{paper}）时，我们指的是使用全部（直接 ∪ 重叠）边，除非另有说明。实际上，我们预计每个使用场景只会使用其中一部分。

4.2 模型卡相关性

其次，在模型卡层面，我们会考察模型卡中提供的基础模型标签，以确定一个模型是否是另一个模型的祖先模型，以及两个模型是否共享同一个祖先模型。我们将此类模型视为相关模型。此外，如果一个模型卡包含指向另一个模型卡的直接链接，则这两个模型也被视为相关。

更正式地说，若满足以下任一条件，则两个模型具有模型相关性（R_{model}）：

（a）显式模型引用 ——一个模型卡明确链接到另一个模型卡；或（b）基础模型继承 ——模型在元数据中声明另一个模型为其基线_模型（或等效项），表明其通过微调、适配器注入、量化或模型合并形成谱系；或（c）共享祖先模型 - 两个模型共享同一个基线_模型。

超链接会按照 https://huggingface.co/{org_id}/{repo_id} 模式解析为规范标识符；而简写引用（例如，用 gpt2 代替 openai-community/gpt2）则使用下载频率统计进行规范化。

4.3 数据卡相关性

最后，在数据集层级，如果两个模型是在同一数据集上训练的，或可推断它们会在同一数据集上运行，则认为这两个模型相关。更正式地说，如果两个模型的模型卡至少引用了一个共同的数据集（通过数据集标签或 URL），R_{dataset} 则这两个模型在数据集层级上相关。数据集标识符从模型卡 README 中提取（例如，`https://huggingface.co/datasets/{org_id}/{repo_id}`），并根据 Hugging Face 官方数据集元数据进行验证[12]，以确保规范对齐。

4.4 模型与表格相关性图

每个表格 t_i 都会继承其关联模型 $M(t_i)$ 的相关性；因此，表格层级的关系定义如下：

$$R_T(t_i, t_j) = R_{\text{paper}}(M(t_i), M(t_j)) \cup R_{\text{model}}(M(t_i), M(t_j)) \cup R_{\text{dataset}}(M(t_i), M(t_j)) \quad (1)$$

不过，该基准的构建方式使不同类型的相关性既可以单独使用，也可以结合使用。随后，我们在构建真实相关性矩阵（各种 R）时将这些定义具体化，详见算法 2。

[12] https://huggingface.co/datasets/librarian-bots/dataset_数据_卡_元数据，版本 250428

算法 2： 表格真实值推断

输入： 表格 T；引文图 G_{cite}；筛选配置 F_{config}
输出： 二元相关性矩阵 $\{R^{paper}_{\mathcal{T}}, R^{model}_{\mathcal{T}}, R^{dataset}_{\mathcal{T}}\}$

函数 ArePapersRelated(m_a, m_b, G_{cite}, F_{config})：
  对每个 $(p_a, p_b) \in GetPapers(m_a) \times GetPapers(m_b)$ **执行**
    $(R_a, R_b) \leftarrow$ GetRefs(p_a, p_b, G_{cite})
    $(R_a, R_b) \leftarrow$ FilterByIntent(R_a, R_b, F_{config}.意图)
    如果 F_{config}.影响 **则**
      $(R_a, R_b) \leftarrow$ FilterByInfluence(R_a, R_b)
    如果 $F_{config}.type$ = '直接' **则**
      如果 $(p_b \in R_a)$或$(p_a \in R_b)$ **则**
        返回 真
    否则如果 $F_{config}.type$ = '重叠' **则**
      如果 $HasOverlap(R_a, R_b)$ **则**
        返回 真
  返回 假
函数模型相关性判断(m_a, m_b)：
  返回 谱系关系(m_a, m_b) 或 共享基础模型(m_a, m_b)
函数数据集相关性判断(m_a, m_b)：
  返回 共享数据集(m_a, m_b)
过程主程序：
  将 $R^{paper}_{\mathcal{T}}, R^{model}_{\mathcal{T}}, R^{dataset}_{\mathcal{T}}$初始化为零矩阵
  对于$(t_a, t_b) \in \mathcal{T} \times \mathcal{T}$ **执行**
    对于 $(m_a, m_b) \in$ 获取模型$(t_a) \times$ 获取模型(t_b) **执行**
      如果 ArePapersRelated($m_a, m_b, G_{cite}, F_{config}$) **则**
        $R^{paper}_{\mathcal{T}}(t_a, t_b) \leftarrow 1$
      如果 模型相关性判断(m_a, m_b)**则**
        $R^{model}_{\mathcal{T}}(t_a, t_b) \leftarrow 1$
      如果 数据集相关性判断(m_a, m_b)**则**
        $R^{dataset}_{\mathcal{T}}(t_a, t_b) \leftarrow 1$
  返回 $\{R^{paper}_{\mathcal{T}}, R^{model}_{\mathcal{T}}, R^{dataset}_{\mathcal{T}}\}$

5 基准统计

本节将分析该基准的特征。

5.1 表格统计

模型级质量筛选。 我们从一个包含超过 110 万个模型及其相关模型卡的模型湖集合开始构建基准。如图 3（上）所示，这些模型卡中约有 68% 非空，至少包含最低限度的文档。其中，约 17% 包含可提取表格，用于总结模型特征或结果。为了创建一个包含学术出版物的高质量模型语料库，我们进一步将范围限定为引用有效研究论文的模型卡。这一步大幅缩小了候选范围，因为许多由社区提交的卡片缺少正式的论文链接。为缓解这一稀疏性，我们整合了其他资源，包括 GitHub README 文件、arXiv 和 Semantic Scholar，以补充缺失的表格或引文信息。经过这项增强后，约有 6 万个模型符合所有质量标准，构成了基准的核心。

表格级重复分析。 图 3 的底部通过展示提取后表格出现的频率，对所得表格语料库进行了分析。我们观察到明显的长尾分布：大多数表格只出现一次，而少数热门或广泛复用的表格则在多个模型卡中反复出现。这

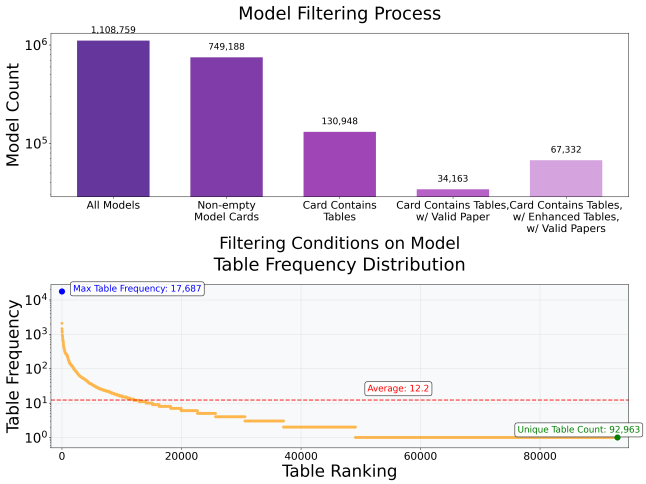


图3：模型湖的模型级与表格级筛选。上：根据模型卡完整性、表格存在性和有效引用逐步筛选模型，最终得到基准子集。下：表格频率分布显示出长尾式复用模式，这促使我们进行去重。

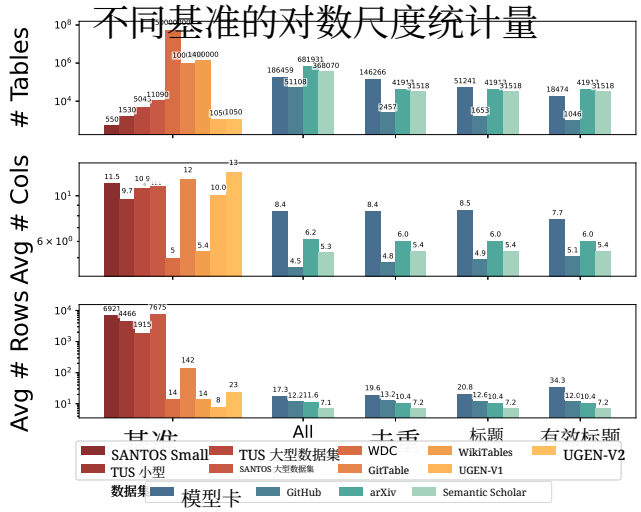


图 4：与数据湖相比，我们的基准包含数量更多、规模更小的表格；这一结构特征源于其学术来源的独特性质。

这与我们的直觉一致：有效的模型卡通常包含一张或几张独特的表格，而偶尔出现的异常则对应于被复用或解析错误的表格。这些证据验证了我们的去重策略，确保基准中保留的表格具有代表性且不存在冗余。

基准筛选概述。 我们的目标是将提取出的表格原始语料库精炼为高质量基准，以便计算表格级和模型级相关性。图 4（右侧的蓝色/绿色色调）总结了数据集内容相对于某些表格筛选条件的变化。

阶段 A - 全部 -> 去重（去重分析）。 删除冗余表格后，模板化或基于模板生成的表格长尾得以消除，最终保留了约 17% 的原始语料库。平均列数几乎没有变化，而平均行数略有增加，这表明重复项主要是较小、重复性较强的表格，而非结构丰富的表格。不同来源的重复程度差异显著：代码资源 GitHub 存在大量重复使用，而我们的两个论文来源 arXiv 和 Semantic Scholar 的冗余则少得多。

阶段 B - 去重 -> 标题（包含引用）。 下一步是将每个表格关联到至少一个文本锚点，例如模型卡标题、GitHub README 标题或论文标题。该约束主要会移除缺少引文或书目上下文的代码仓库表格，同时保留大多数论文表格。从概念上看，语料库由原始结构化成果转变为以引用为锚点的表格，从而支持后续计算论文级和模型级关系。

阶段 C - 标题 -> 有效标题（学术验证）。 随后，使用外部学术索引（Semantic Scholar）对标题进行验证，以确认每个标题都对应一个可核验的研究成果。经过此次筛选后，基准稳定在约 10^5 个表格。在这一子集中，通过验证的模型卡片表格明显规模更大（行数更多），但更为狭窄（属性更少），因为规模更大的性能表占据主导地位。大约三分之二的模型卡标题和约三分之一的 GitHub 引用未能通过匹配，而几乎所有论文标题都如预期般顺利解析。由此保留下来的语料库体现了具有学术依据且结构一致的内容，适用于定量相关性分析。

与现有语料库和基准的比较。 图4（左侧橙色/红色阴影部分）展示了若干热门表格语料库和一些数据湖基准（后者包含真实标注）。与经典数据湖基准（如 SAN- TOS [27] 和 TUS [43]）以及较新的大语言模型生成的基准 UGEN V1/2 [46]相比，我们的基准呈现出独特的结构和语义特征。我们收录的表格数量多得多，但规模较小（与较小的 UGEN 表格规模相当）。尽管总体统计数据有所不同，这两类基准仍具有互补性。请注意，数据湖基准要么是从开放数据中人工整理而来，因此已知真实标签之间的相关性（TUS/SANTOS），要么由大语言模型生成（UGEN）。正如下文所述，这些基准的差异更多体现在真实值的特征（相关性分布）上，见第 5.2节。传统数据湖捕捉大规模的结构多样性，而我们的基准则强调具有学术结构的精确性。二者共同构成了更广阔的研究图景，可用于探索异构领域中的表格发现、集成和评估方法。相比之下，规模更大的（未标注）网络语料库（WDC、WikiTables、GitTables）包含的表格数量要多得多。除 GitTables 外，这些语料库中的表格通常规模较小，因为它们来自网页和大众协作页面。

5.2 表相关性统计

我们已经考察了模型和表格的过滤质量（图3、4），确保每个保留的表格都关联到一篇有效的

论文。在本节中，我们将转而讨论根据三种关联类型得出的真实关联度统计。

论文相关性真实标注密度。我们首先量化八个不同子集中的表格论文相关性（R_{paper}）密度 R_{paper}。表 2。原始重叠图连接了超过 40% 的全部表格对，反映出广泛但噪声较大的关联。应用影响力或意图筛选器后，这一密度降至 10–25%；结合二者后，则形成一个紧凑的、约占 8% 的高置信度核心。相比之下，直接引文图仍然稀疏得多（约为 3–8%），表明其精确率更高，但覆盖范围更窄。这些统计结果证实，我们的真实图保留了足够的连通性，能够支持有意义的发现任务，同时不会退化为不切实际的稠密结构。密度上的差异凸显了该基准的两种互补能力。基于重叠的图能够恢复间接的语义关系，而直接图则表示明确的引用沿袭。二者共同构成精确率—召回率权衡：重叠关系带来更广的覆盖和更高的召回率；直接引文则提供精确率和可靠性。

表2：不同引文类型下的论文相关性密度。应用影响力和意图筛选器会使图变得稀疏，而基于重叠的方法能够捕捉间接关系，从而形成密度高得多的图。

引文类型	非零边	密度
重叠	3.71×10^9	43。03%
重叠（意图）	2.23×10^9	25。83%
重叠（影响力）	8.59×10^8	9。94%
重叠（意图 ∩ 影响力）	6.95×10^8	8。04%
直接	7.06×10^8	8.17%
直接（意图）	4.98×10^8	5.76%
直接（有影响力）	3.01×10^8	3.48%
直接（意图 ∩ 有影响力）	2.67×10^8	3.09%
模型	1.71×10^7	0.19%
数据集	3.51×10^7	0.41%
全部（直接 ∪模型 ∪ 数据集）	7.18×10^8	8。31 %

模型和数据集密度模式。 表格 2底部展示了表格图中模型和数据集相关性的密度。模型相关性最为稀疏，这符合预期，而且在语义上最为精确。数据集相关性也较为稀疏，但代表了另一种相关性语义，是否纳入应视具体任务而定。

表格相关性分布。 采用数据湖中常见的以表格为查询的范式 [42]，我们进一步分析每个查询对应的相关表格数量分布，即有多少个相关表格

每个查询表有（图5）。与 SANTOS [27] 和 TUS [43]，等现有基准不同，这些基准定义了一组固定的查询，而我们的基准将每个表格都视为潜在的查询。[13] 这种设计大幅扩展了查询空间，但这并不意味着底层分布截然不同。事实上，我们的

[13]排除 UGEN，因为每个查询都恰好有 10 个相关表格。

基准呈现出更广泛、变化更大的长尾模式，相关表格数量从寥寥数个到一千多个不等。这种模式源于枢纽效应：被高度引用的论文或广泛复用的基础模型会形成连接密集的簇。由此产生的差异反映了科学表格生态系统真实且异质的特性，为评估检索与发现方法提供了具有挑战性且具代表性的真实标注。

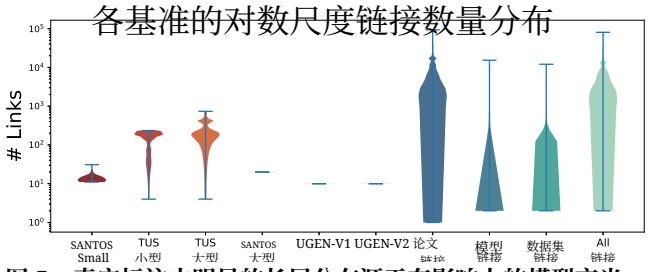


图 5：真实标注中明显的长尾分布源于有影响力的模型充当枢纽，并与许多模型卡相连。

6 使用 ModelTables 评估表搜索

现在，我们将介绍基于基准开展的详细实验结果。在本研究中，我们评估现有的相关表格搜索方法在多大程度上能够根据我们的任一真实标注指标（模型、数据或论文）找到相关表格。

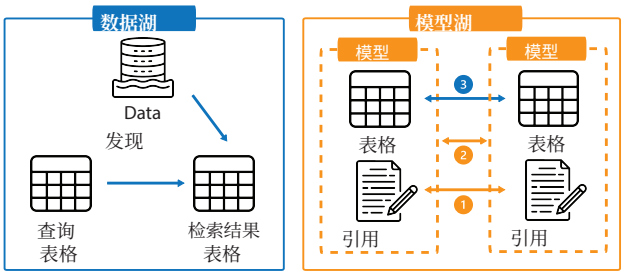


图 6：在模型湖背景下界定表格发现任务。为了赋予传统表格搜索（左）实际意义，我们利用相关论文、模型和表格之间丰富的联系，建立相关性的真实标注（右）。

本节将在我们新构建的 ModelTables 基准上，对多种表格发现方法进行全面评估。我们考察现有技术在模型表上的表现，包括先进的表格发现技术，以及传统的基于嵌入和信息检索中的元数据驱动的检索方法。

6.1 实验方法

我们在自己的基准上评估最先进的表搜索和检索方法。我们考察的数据发现方法包括

表 3：表格发现与检索方法的实现和设置对比

	表发现方法			检索方法		
方法	关键词搜索	可连接搜索	可合并搜索	稠密检索	稀疏检索	混合检索
内容	表格	表格	表格	表格	元数据	元数据
Tool	Blend [15]	Blend [15]	Starmie [17]	FAISS [13]	Pyserini [34]	Pyserini [34] +FAISS [13]
表头	Yes	No	No	Yes	-	-

可连接和可并集的表搜索。我们还考察最先进的表检索方法。表3总结了区分这些方法的关键实现特征。

关键词搜索。 我们使用 Blend 对关键词搜索的实现 [15]。Blend 是一个较新的统一表格发现系统，包含一系列发现算子，其中包括关键词搜索、连接、相关性和并集发现。对于关键词搜索，我们通过将查询表的表头词元与候选表的表头和单元格进行匹配来检索表格。如果查询词元包含在候选表的内容中，我们就认为该表格相关，并根据命中次数对候选表进行排序。

连接搜索。 我们使用 Blend 对连接搜索的实现 [15]。给定一个查询表，我们搜索在最右列上可进行连接的表格（该列通常是模型湖表中的键列）。遵循相关研究的做法，我们在此搜索中不使用表头，因为属性名称可能不同，但仍然可以连接。

并集搜索。 我们使用 Starmie 对并集搜索的实现 [43]。与仅考虑表格值精确匹配的 Blend 不同，Starmie 考虑语义匹配，在此任务上的表现优于 Blend。

稠密检索。 我们创建一种表嵌入，将每个表格（值和表头）编码为一个向量，并使用预训练的 Sentence-BERT 编码器 [50]。随后根据候选表嵌入的余弦相似度对其排序，并使用 FAISS [13] 高效完成这一过程。

稀疏检索。 稀疏检索是一种传统的信息检索技术，依赖术语匹配，并已通过匹配描述表格的元数据用于表检索 [40, 56]。此类元数据检索利用每个表格周围的文本上下文，而非其内部结构。我们的预处理会汇总原始 README 文本（针对 Markdown 表格），或图注与正文提及（针对 arXiv 表格），为每个表格构建局部的“上下文内”元数据语料库。值得注意的是，当一个表格关联多条元数据文本（因重复所致）时，我们会将这些文本拼接起来，创建用于索引的增强语料库。我们将这些元数据文本序列化为提示，以生成查询。对于稀疏检索，我们使用 Pyserini [34] 稀疏词项索引，并将截断长度设为 1024 个子句；对于混合检索，则先通过稀疏搜索检索排名前 100 的候选项，再使用密集型 Sentence-BERT 进行重新排序。

混合检索。 混合检索以级联方式结合稀疏检索与基于元数据的稠密检索。具体而言，我们首先应用稀疏元数据检索，获取排名前 100 的候选项，然后使用稠密检索从中选出排名第一的结果。这种双重方法使我们能够将纯稀疏检索与级联混合方法的优势进行比较。

6.2 评估结果

设置。 在评估中，我们使用原始 ModelTable 语料库作为默认的检索候选池。在增强消融研究中，我们将这些表格与转置增强和表头增强（第 3.3 节）合并。因此，对于这些消融设置中的每个查询表，我们会（并行）检索该表、其转置表和表头增强版本。如果任一检索返回的排名第一结果对应于正确的真实表，则视为检索成功。本研究采用直接论文相关性。这种乐观的评估设置旨在估算当前每种最先进方法所能达到的性能上限。

研究发现。 本节介绍不同方法的基线检索性能。其目标是为未来的改进建立参考基准，并了解不同的数据特征如何影响检索效果。

F1值：尽管所有现代方法的表现都显著优于关键词搜索，但除并集搜索外，信息检索方法的表现均优于数据湖方法。 我们的研究发现（表 4）表明，现代语义检索方法——包括基于表内容、上下文元数据和模式感知的方法——在所有定义的真实值（GT）上，表现始终显著优于传统关键词搜索。然而，这些方法的效果各有差异，具体取决于每种方法所优先采用的语义信号。

在检索方法中，表格稠密检索被证明非常有效，在论文真实标注上取得了顶尖性能，并且总体结果最佳，这凸显了捕捉表格语义的重要性。与此同时，上下文元数据对于特定任务至关重要：混合检索利用原始表数据之外的重要上下文信息，在模型真实标注和数据集真实标注上均表现出色。可并集方法 Starmie 表现强劲，在所有真实标注上均领先于表发现方法。值得注意的是，Starmie 在论文真实标注和全部真实标注上排名第二，表明可并集性是直接论文相关性的强信号。然而，在考察总体表现或其他特定真实标注时，它通常不如稠密检索和混合检索，这表明整合这些语义信号仍有提升空间。重要的是，正如我们稍后展示的，这些方法检索到的表格各不相同，进一步表明有必要开展研究，结合它们各自的优势。

表4：不同真实值（GT）下检索和表格发现方法的精确率@1

检索方法					表格发现方法				
方法 / 真值	论文	模型数据集		All	方法 / 真值	论文	模型数据集		All
稠密检索	**0.6651**	0.4175	0.4143	**0.6672**	关键词搜索	0.2043	0.0472	0.0460	0.2060
稀疏检索	0.5131	0.3724	0.3749	0.5157	可连接搜索	0.2742	0.0694	0.0723	0.2755
混合检索	0.5410	**0.4572**	**0.4592**	0.5428	可合并搜索	*0.5465*	0.3133	0.3068	*0.5487*

F2：性能差异显著，源于表格来源质量的不同。表格来源会显著影响其风格和数据质量。例如，来自 GitHub 和模型卡的表格通常包含开发者整理的丰富元数据，而来自学术论文、arXiv 和 Semantic Scholar 的表格在结构上则更加多样。为分析这一点，我们评估了可并集搜索在各个来源上的单独表现及其组合表现（表5）。

我们的分析表明，不同来源的精确率差异显著。模型卡和 GitHub 表格的精确率始终较高，这可能得益于其经过精心整理的 Markdown 格式。请注意，Semantic Scholar 表格是经大语言模型优化的纯文本，而非结构丰富的 HTML 或 Markdown 表格。因此，这些表格的噪声更大，严重影响了检索质量。

我们对组合来源进行的实验进一步证实了来源特征影响这一原理，并凸显了数据同质性对于实现高精确率至关重要。在模型卡和 GitHub（M+G）子集中，表格采用一致的 Markdown 格式，因此性能最佳。随着结构较弱的 arXiv 表格被引入（在 M+G+A 中），准确率逐步下降；加入更多样化的 Semantic Scholar 表格后（在 M+G+A+S 中），准确率进一步降低。

F3：领域特定增强提升表格的可搜索性。为改进检索（尤其是数值表格的检索，它们通常缺乏明确的语义上下文，且布局各异），我们分别评估了两种数据增强策略：添加语义上下文（表头增强）和规范化布局（转置增强）。请注意，基于元数据的检索方法不纳入此分析，因为它们无法从表格增强中受益。可合并搜索也不纳入其中，因为 Starmie 的运行时间取决于列数，而列数在转置增强下呈爆炸式增长。我们的分析（表格 6）展示了每种增强的效果。

添加语义上下文被证明非常有效，因为它提高了与所有真实值相关性（论文、模型和数据集）相关的精确率。这证实了明确数值单元格的隐含含义对于模型湖表检索非常有价值。相比之下，转置增强始终只能略微提升性能。这表明，对于模型湖表而言，最佳的增强策略是丰富表格现有的语义。

F4：在不同真实标注下，稳健性与性能一致性均得以保持。我们的基准允许使用不同的引用语义来确定论文之间的关联性（表2）。表7 展示了并集搜索在八种不同引用定义下的精确率。当应用更严格的真实值标准时，精确率会如预期般下降（即引用关系更少），这是因为真实标注变得更加稀疏。使用论文真实标注时，这种反向关系尤为明显：随着引用标准日益严格，精确率随之下降。有趣的是，比较不同的 GT 时发现，基于重叠定义的 GT 始终比基于直接引用标准的 GT 精确率更高，这表明重叠天然涵盖了范围更广、密度更高的相关连接。

这些发现表明，尽管检索性能指标对真实标注本身的定义和统计属性高度敏感，但即使真实值标准变得更加严格，各种方法仍表现一致。这凸显了理解 GT 选择如何影响报告的性能的重要性。

F5：结构增强与语义增强相结合，能够有效提升检索鲁棒性。除语义增强和引用增强外，我们还研究了表结构扰动的影响，包括列打乱和行打乱。由于 Starmie 的编码器专为结构鲁棒性而设计，这些增强仅使用 Starmie 进行评估。推理阶段进行列打乱可带来最大的增益，表明增强表结构是提升异构模型湖中检索鲁棒性的一种实用策略。

我们观察到，所有结构增强技巧都能提升检索性能，这证实了不同作者和来源的表格样式存在显著差异。列打乱带来的提升最大，表明列的顺序尤其随意——在采用半结构化格式的学术表格中更是如此，因为这类表格的可并集性很难检测。结构增强，尤其是列级增强，有助于编码器关注语义和模式相似性，而非布局，从而在异构模型湖中实现更稳健的检索。

表格 5：不同 GT 上各表格资源并集搜索的精确率@1。

资源 / GT	论文	模型	数据集	All
模型卡 (M)	0.7857	0.2599	0.2142	0.7873
GitHub (G)	0.8177	0.7381	0.7366	0.8171
arXiv (A)	0.4504	0.3545	0.3607	0.4517
Semantic Scholar (SS)	0.3059	0.2630	0.2637	0.3062
M+G	0.7801	0.2766	0.2358	0.7816
M+G+A	0.5465	0.3133	0.3068	0.5487
M+G+A+SS	0.4404	0.2700	0.2657	0.4423

表6：不同 GT 上增强表格的方法精确率@1。

方法 / GT	表头到单元格增强				转置增强				两者增强			
	论文	模型	数据集	All	论文	模型	数据集	All	论文	模型	数据集	All
稠密检索	0.7778	0.5121	0.5042	0.7798	0.7602	0.4950	0.4866	0.7623	0.8142	0.5459	0.5358	0.8160
关键词搜索	0.2738	0.0512	0.0502	0.2761	0.2989	0.0785	0.0782	0.3012	0.4653	0.2365	0.2473	0.4688
可连接搜索	0.4569	0.1541	0.1555	0.4593	0.2751	0.0527	0.0592	0.2776	0.5056	0.1617	0.1651	0.5084

表7：基于论文真实标注的引用扩充可合并搜索精确率@1。

引文类型	直接	重叠
All	0.5465	0.7487
意图	0.5278	0.6471
有影响力	0.5097	0.5533
意图 ∩ 有影响力	0.5064	0.5382

表格 8：Starmie 在不同真值上使用结构增强的精确率@1。

增强 / 真值	论文	模型	数据集	All
Base	0.4119	0.1899	0.1806	0.4133
删除单元格	0.4607	0.2323	0.2240	0.4627
打乱列	**0.5465**	**0.3133**	**0.3068**	**0.5487**
打乱行	0.4419	0.2031	0.1979	0.4436

6.3 示例研究

即使对于简单的 precision1 任务，已知检索方法在 ModelTables 上的性能也都不理想。为了更清楚地说明原因，我们给出两个示例：一个是使用结构化技术（可合并搜索）检索到的表格，另一个是使用表嵌入（稠密检索）检索到的表格。我们借助这些示例直观地说明，为什么我们认为本研究凸显了谨慎结合结构信号与语义信号的必要性，以改进相关表搜索。

示例 I：可合并搜索示例。这里我们演示如何对 GLUE 风格性能表使用可合并搜索，展示该搜索检索描述相关模型的表格的能力。图7 (a) 展示了一个 BERT 查询表，以及使用 Starmie（并集表搜索）检索到的排名前 2 的表格。这些搜索结果显然相关，能够直接并排比较基线模型和高级模型（例如 BERT [11], RoBERTa [37],MUPPET [1], Uni-Perceive [63]），并且涵盖一组一致的自然语言处理任务和评估指标，例如多体裁自然语言推理、问答自然语言推理、MRPC、SST-2 和 SQuAD [49]。这些表格在结构和语义上具有明显的对齐性，报告了重叠基准上的结果，因此可以合并。不过，对于分析模型行为的数据科学家而言，这些表格也显然具有相关性。

我们的观察凸显了传统文本信息检索方法的局限性：这类方法只能回答诸如“给定任务有哪些可用模型？”之类的粗粒度问题，无法在指标级别进行细粒度性能比较。通过表搜索，我们可以检索相关且可整合的表格，从而扩展仅凭查询表所能进行的分析。这种方法使用户能够找到有助于他们做出精准且结合上下文的决策的表格，而这些决策基于多个模型和基准的详细性能数据。

示例 II：稠密检索示例。如图7(b) 所示，稠密检索找到了这些表格：它们仍以 BERT 变体和 GLUE、SQuAD 等关键自然语言处理基准为中心，但与基于并集的结果相比，其模式和报告风格的异质性要大得多。与并集搜索找到的模式对齐程度更高的表格不同，稠密检索往往会返回列组织方式各异、指标更多或格式不太常见的表格，这反映了作者呈现实验结果的多种方式。

这种灵活性可以被视为一项重要优势：稠密检索能够连接语义相关、但结构并不相同或采用不同约定的表格。因此，即使列名、任务集或布局与查询存在差异，它仍能成功检索到相关数据。对于科学语料库而言，这一能力尤其宝贵，因为其中的报告标准差异很大，元数据也可能存在噪声或不完整。

然而，稠密检索也并非没有局限。由于它依赖语义相似性，因此往往会检索到基线或基准高度重叠的表格——尤其是聚焦于“BERT-base”结果的表格——而不是涵盖更多模型变体或实验设置的表格。这可能导致检索到的表格集中报告相同的基线，从而遗漏那些虽然在结构上可以合并、但涵盖更广泛模型或替代方法的表格。相比之下，基于合并的方法更有可能呈现多样的模型变体和实验配置，从而支持跨方法进行更广泛的比较。

总之，稠密检索通过弥合模式对齐方面的差距，并找出结构多样且语义相关的表格，有效地补充了基于并集的方法。然而，它在促进方法或模型多样性方面效果较差，往往优先选择在基线和基准方面存在显著重叠的表格。这表明，如何结合两种方法的优势以提升模型湖性能，仍是亟待解决的研究挑战。

7 应用

除了相关表搜索之外，我们的基准还有许多应用。下面仅介绍其中几项。

模型搜索 不同于以往的表格语料库，ModelTables 可用于模型理解。一个可能的研究方向是

查询表：BERT - HTML

任务	开发集				
	MNLI-m (准确率)	QNLI (准确率)	MRPC (准确率)	SST-2 (准确率)	SQuAD (F1)
BERT_BASE	84.4	88.4	86.7	92.7	88.5
无下一句预测	83.9	84.9	86.5	92.6	87.9
从左到右且无下一句预测	82.1	84.3	77.5	92.1	77.8
+ 双向长短期记忆网络	82.1	84.1	75.7	91.6	84.9

检索结果排名第 1：Muppet - HTML （a）可合并搜索

	GLUE						MRC
	MNLI	QQP	RTE	QNLI	MRPC	SST-2	SQuAD
RoBERTa-B	87.6	91.9	78.7	92.8	90.2	94.8	82.6
+ MUPPET'	88.1	91.9	87.8	93.3	91.7	96.7	86.6
RoBERTa-L	90.2	92.2	88.1	94.7	90.9	96.4	88.7
…	…	…	…	…	…	…	…

检索结果排名第 1：PERT - HTML （b）稠密检索

系统	SQuAD		SQuAD 2.0		MNLI	SST-2	CoLA	MRPC
	EM	F1	EM	F1	Acc	Acc	M.C.	Acc
BERT_{base}	80.8	88.5	-	-	84.4	92.7	60.6	86.7
BERT_{base} †	81.2	88.5	72.4	75.4	84.4	92.6	59.3	86
RoBERTa_{base}	-	90.4	-	79.1	84.7	92.5	-	-
…	…	…	…	…	…	…	…	…

检索结果排名第 2：Uni-Perceive - HTML (a) 可合并搜索

Task	GLUE					
	MNLI (准确率)	QNLI (准确率)	QQP (F1)	RTE (准确率)	SST-2 (准确率)	MRPC (F1)
预训练语言模型[3] 未微调	49.4	50.7	46.6	53.8	70.6	44.2
BERTBASE[70]	84.6	92.7	71.2	66.4	93.5	88.9
我们的方法未微调	49.6	51	53.6	55.6	70.6	76.1
…	…	…	…	…	…	…

检索结果排名第 2：LTG-BERT - HTML (b) 稠密检索

模型（变体）	GLUE					HANS	边缘探测	BLiMP
	MNLI	MRPC	QNLI	SST-2	平均值			
维基百科 + BookCorpus（3000M 个词）								
BERT_{基线, cased} †	84.4	86.7	88.4	92.7	88.1	69	93.9	84.2
BERT_{基线, cased}（我们的评估）	83.6+-0.2	84.6+-0.5	90.8+-0.1	91.9+-0.4	87.8+-0.3	61.8+-1.5	93.8+-0.2	84.2
…	…	…	…	…	…	…	…	…

图 7：使用（a）基于并集的方法和（b）稠密检索，为 BERT 查询进行表格发现的示例。（a）基于并集的检索会返回模式对齐、GLUE [53] 和 SQuAD [49] 结果报告一致的表格。（b）稠密检索会找到更多样、半结构化的表格，反映出作者在结果报告和布局方面的不同做法。

探究表格相关性是否可以作为模型关联性的代理指标。目前，模型湖依赖基于名称和元数据的关键词搜索。语义搜索才刚刚兴起，并依赖于针对开放模型的大语言模型问答。我们假设，描述相关模型的表格具有较高的结构相似性和语义相似性，因此可以将表格发现作为识别相关模型的代理方法。研究表明，科学表格包含丰富的语义，有助于推理和检索 [10, 31, 39, 55, 60]。验证这一假设需要：（1）稳健的表检索方法；以及（2）具有原则性的模型关联性真实定义。

模型表集成虽然集成一组数据湖表的问题已得到研究 [28], , 但集成不一致的数据湖表（包括含有许多数值的表格）这一问题尚未得到足够关注。借助 ModelTables，我们可以识别出一组关于某个模型的表格，并将其归类为配置表、性能表和其他类型的表格。在每种类型的表格中，我们可以识别表格间的不一致之处，并使用集成技术解决这些不一致问题，从而定义出最佳集成方案。我们推测，此类研究不仅可能对数据集成研究产生影响，也可能对模型卡验证和生成研究产生影响 [36]。

模型理解 模型湖研究的一大终极目标，是了解哪种模型最适合某项任务 [45]。为此，我们推测，通过对模型表进行有原则的搜索，我们或许能够了解哪些模型在不同基准上的表现最佳，并综合这些丰富的结构化信息，从而更好地对模型进行问答。

8 结论与未来工作

我们介绍了一个新的基准 ModelTables，并提出了稳健且可重复的方法来生成该基准及其真实标注。该基准基于 2025年1月 对 Huggingface 及相关资源进行的爬取而构建。如图 8 ，随着时间推移，表格和模型的数量呈指数级增长，这将使生成规模显著更大的

我们利用可复现的开放基准创建流程和代码构建了 ModelTables V2。我们介绍了基准表格的特征，并展示了从模型开发者创建的真实可靠信号中提取的真实标签呈现出真实可信的特征（例如枢纽热门模型和倾斜的相关性分布），这些特征在所调查的所有表格基准中独具一格（这些基准使用人工标注或大语言模型生成的标签）。我们介绍了该基准的一个综合应用案例：评估模型表上的相关表格搜索方法；与网页表格或数据湖不同，在模型表中，相关性不能仅依据结构来定义 [51]。我们展示了该基准如何帮助洞察研究空白和创新方向。我们还推测了该基准在表格理解和模型理解方面的其他若干应用。

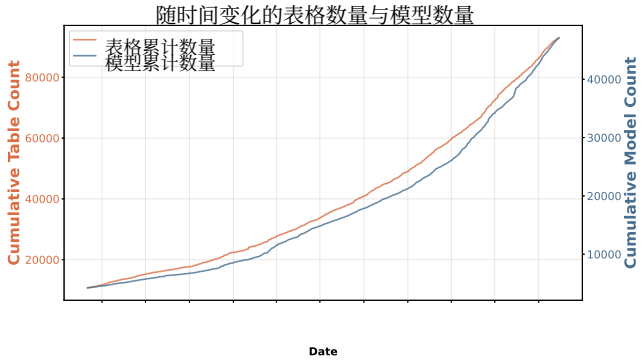


图 8：Huggingface 模型和表格随时间变化的累计数量。

致谢

我们感谢加拿大卓越研究主席计划（CERC）的支持。Nous remercions le Chaires d'excellence en recherche du Canada (CERC) de son soutien.

参考文献

[1] Armen Aghajanyan、Anchit Gupta、Akshat Shrivastava、Xilun Chen、Luke Zettlemoyer 和 Sonal Gupta。2021。MUPPET：通过预微调实现大规模多任务表示。载于《2021年自然语言处理实证方法会议论文集》。5799–5811。[2] Mehdi Akbarian Rastaghi、Ehsan Kamalloo 和 Davood Rafiei。2022。探究预训练语言模型在实体匹配任务中的鲁棒性。载于《第31届ACM信息与知识管理国际会议论文集》（美国佐治亚州亚特兰大）（CIKM '22）。计算机协会，纽约，美国，3786 3790 10 1145 3511808 3557673 – 。https://doi.org/ 。/ 。[3] Chandra Sekhar Bhagavatula、Thanapon Noraset 和 Doug Downey。2013。维基百科表格的探索与挖掘方法。载于 SIGKDD 交互式数据探索与分析研讨会 IDEA@KDD。ACM，18–26。https://doi.org/ 10 1145 2501511 2501516 。/ 。[4] Alex Bogatu 、 Alvaro A. A. Fernandes 、 Norman W. Paton 和 Nikolaos Konstantinou。2020。数据湖中的数据集发现。载于 ICDE。709–720。[5] Peter Buneman、Dennis Dosso、Matteo Lissandrini 和 Gianmaria Silvello。2022。数据引用与引文图。Quantitative Science Studies 2, 4 (2022), 1399–1422。[6] Michael J Cafarella、Alon Halevy、Daisy Zhe Wang、Eugene Wu 和 Yang Zhang。2008。Webtables：探索网络表格的潜力。Proceedings of the VLDB Endowment 1, 1 (2008), 538–549。[7] Adriane Chapman、Elena Simperl、Laura Koesten、George Konstantinidis、Luis-Daniel Ibáñez、Emilia Kacprzak 和 Paul Groth。2020。数据集搜索：综述。VLDB J. 29, 1 (2020), 251–272。https://doi.org/10.1007/S00778-019-00564-X[8] Christina Christodoulakis、Eric B Munson、Moshe Gabel、Angela Demke Brown 和 Renée J Miller。2020。Pytheas：CSV 文件中的基于模式的表格发现。Proceedings of the VLDB Endowment 13, 12 (2020), 2075–2089。[9] Arman Cohan、Waleed Ammar、Madeleine van Zuylen 和 Field Cady。2019。科学出版物中引用意图分类的结构支架。载于《2019年北美计算语言学协会年会：人类语言技术论文集》第1卷（长篇与短篇论文）。3586–3596。[10] Harsh Desai、Pratik Kayal 和 Mayank Singh。2021。TabLeX：用于科学表格结构与内容信息抽取的基准数据集。载于《文档分析与识别——ICDAR 2021：第16届国际会议，瑞士洛桑，2021年9月5日至10日，论文集》第II部分第16卷。Springer，554–569。[11] Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019。BERT：用于语言理解的深度双向 Transformer 预训练。载于《2019年北美计算语言学协会年会：人类语言技术论文集》第1卷（长篇与短篇论文）。4171–4186。[12] Yuyang Dong、Chuan Xiao、Takuma Nozawa、Masafumi Enomoto 和 Masafumi Oyamada。2023。DeepJoin：利用预训练语言模型进行可连接表格发现。Proc. VLDB Endow. 16, 10 (2023), 2458–2470。https://doi.org/10.14778/ 3603581.3603587[13] Matthijs Douze、Alexandr Guzhva、Chengqi Deng、Jeff Johnson、Gergely Szilvasy、Pierre-Emmanuel Mazaré、Maria Lomeli、Lucas Hosseini 和 Hervé Jégou。2024。Faiss库。（2024）。arXiv:2401.08281 [cs.LG][14] Julian Eberius、Katrin Braunschweig、Markus Hentsch、Maik Thiele、Ahmad Ahmadov 和 Wolfgang Lehner。2015。构建德累斯顿网页表格语料库：一种分类方法。载于《2015年IEEE/ACM第二届大数据计算国际研讨会（BDC）》。IEEE，41–50。[15] Mahdi Esmailoghli、Christoph Schnell、Renée J Miller 和 Ziawasch Abedjan。2025。BLEND：统一数据发现系统。载于《2025年IEEE第41届数据工程国际会议（ICDE）》。IEEE计算机学会，737–750。[16] Grace Fan、Roee Shraga 和 Renée J Miller。2024。Gen-T：数据湖中的表格回收。载于《2024年IEEE第40届数据工程国际会议（ICDE）》。IEEE，3532–3545。[17] Grace Fan、Jin Wang、Yuliang Li、Dan Zhang 和 Renée J. Miller。2023。利用基于上下文的列表示学习，从数据湖中发现语义相关的数据集。PVDLB 16, 7 (2023), 1726–1739。[18] Suzanne Fricke。2018。Semantic Scholar。《Journal of the Medical Library Association: JMLA》106, 1 (2018), 145。[19] Moncef Garouani、Franck Ravat 和 Nathalie Vallès-Parlangeau。2024。模型湖：机器学习模型管理与治理的新替代方案。载于《Web信息系统工程——WISE 2024——第25届国际会议，卡塔尔多哈，2024年12月2日至5日，论文集》第IV部分（《计算机科学讲义》），Mahmoud Barhamgi、Hua Wang 和 Xin Wang（编），第15439卷。Springer，133–144。https://doi.org/10.1007/978-981-96-0573-6_10[20] Timnit Gebru、Jamie Morgenstern、Briana Vecchione、Jennifer Wortman Vaughan、Hanna Wallach、Hal Daumé III 和 Kate Crawford。2021。数据集数据表。Commun. ACM 64, 12 (2021), 86–92。[21] Mazhar Hameed、Gerardo Vitagliano、Fabian Panse 和 Felix Naumann。2025。使用 TASHEEH 修复原始数据文件。SIGMOD Rec. 54, 1 (2025), 90–99。https://doi.org/10.1145/3733620.3733639

[22] Jilin He、Ekaterina Borisova 和 Georg Rehm。2024。《迈向学术出版物表格类型的新型分类》。载于自然科学语言处理与研究知识图谱国际研讨会。Springer，31–48。[23] Ginny Hendricks、Dominika Tkaczyk、Jennifer Lin 和 Patricia Feeney。2020。《Crossref：社区所有学术元数据的可持续来源》。Quantitative Science Studies 1，1（2020），414–427。[24] Kevin Zeng Hu、Snehalkumar (Neil) S. Gaikwad、Madelon Hulsebos、Michiel A. Bakker、Emanuel Zgraggen、César A. Hidalgo、Tim Kraska、Guoliang Li、Arvind Satyanarayan 和 Çagatay Demiralp。2019。《VizNet：迈向大规模可视化学习与基准测试资源库》。载于 CHI。ACM，662。https: //doi.org/ 10 1145 3290605 3300892 . / . Madelon Hulsebos、Çağatay Demiralp 和 Paul Groth。2021。《Gittables：大型关系表语料库》。arXiv 预印本 arXiv:2106.07258（2021）。[26] Madelon Hulsebos、Çagatay Demiralp 和 Paul Groth。2023。《Gittables：大型关系表语料库》。Proceedings of the ACM on Management of Data 1，1（2023），1–17。[27] Aamod Khatiwada、Grace Fan、Roee Shraga、Zixuan Chen、Wolfgang Gatterbauer、Renée J. Miller 和 Mirek Riedewald。2023。《SANTOS：基于关系的语义表并集搜索》。载于即将在 SIGMOD Conference 上发表。ACM。https://arxiv.org/pdf/2209.13589.pdf[28] Aamod Khatiwada、Roee Shraga、Wolfgang Gatterbauer 和 Renée J Miller。2022。《数据湖表集成》。Proceedings of the VLDB Endowment 16，4（2022），932–945。[29] Seongchan Kim、Keejun Han、Soon Young Kim 和 Ying Liu。2012。《数字图书馆中的科学表类型分类》。载于 Proceedings of the 2012 ACM symposium on Document engineering。133–136。[30] Keti Korini、Ralph Peeters 和 Christian Bizer。2022。《SOTAB：WDC Schema.org 表标注基准》。载于 CEUR Workshop Proceedings，第3320卷。RWTH Aachen，14–19。[31] Benno Kruit、Hongyu He 和 Jacopo Urbani。2020。《Tab2know：从科学论文中的表格构建知识库》。载于国际语义网会议。Springer，349–365。[32] Oliver Lehmberg、Dominique Ritze、Robert Meusel 和 Christian Bizer。2016。《大型公开网页表格语料库：包含时间和上下文元数据》。载于 Proceedings of the 25th international conference companion on world wide web。75–76。[33] Quentin Lhoest、Albert Villanova Del Moral、Yacine Jernite、Abhishek Thakur、Patrick Von Platen、Suraj Patil、Julien Chaumond、Mariama Drame、Julien Plu、Lewis Tunstall 等。2021。《Datasets：自然语言处理社区库》。arXiv 预印本 arXiv:2109.02846（2021）。[34] Jimmy Lin、Xueguang Ma、Sheng-Chieh Lin、Jheng-Hong Yang、Ronak Pradeep 和 Rodrigo Nogueira。2021。《Pyserini：用于可复现信息检索研究的 Python 工具包，支持稀疏和稠密表示》。载于 Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)。2356–2362。[35] Zihang Lin、Yian Yin、Lu Liu 和 Dashun Wang。2023。《SciSciNet：面向科学学研究的大规模开放数据湖》。Scientific Data 10，1（2023），315。[36] Jiarui Liu、Wenkai Li、Zhijing Jin 和 Mona T. Diab。2024。《模型卡和数据卡自动生成：迈向负责任的人工智能》。载于 ACL，Kevin Duh、Helena Gómez-Adorno 和 Steven Bethard（编）。计算语言学协会，1975–1997。https://doi.org/10.18653/V1/2024.NAACL-LONG.110[37] Yinhan Liu、Myle Ott、Naman Goyal、Jingfei Du、Mandar Joshi、Danqi Chen、Omer Levy、Mike Lewis、Luke Zettlemoyer 和 Veselin Stoyanov。2019。《RoBERTa：一种稳健优化的 BERT 预训练方法》。arXiv 预印本 arXiv:1907.11692（2019）。[38] Kyle Lo、Lucy Lu Wang、Mark Neumann、Rodney Kinney 和 Daniel Weld。2020。《S2ORC：Semantic Scholar 开放研究语料库》。载于 Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics。计算语言学协会，线上，4969–4983。https://doi.org/10.18653/v1/2020. acl-main.447[39] Xinyuan Lu、Liangming Pan、Qian Liu、Preslav Nakov 和 Min-Yen Kan。 [无日期]。《SCITAB：面向科学表格组合推理与声明验证的挑战性基准》。载于 2023 年自然语言处理实证方法会议。[40] Joel Mackenzie、Andrew Trotman 和 Jimmy Lin。2023。《针对学习型稀疏表示的高效逐文档和逐得分查询评估》。ACM Transactions on Information Systems 41，4（2023），1–28。[41] Margaret Mitchell、Simone Wu、Andrew Zaldivar、Parker Barnes、Lucy Vasserman、Ben Hutchinson、Elena Spitzer、Inioluwa Deborah Raji 和 Timnit Gebru。2019。《用于模型报告的模型卡》。载于公平性、问责制与透明度会议论文集（美国佐治亚州亚特兰大）（FAT* '19）。计算机协会，纽约，NY，美国，220–229。https://doi.org/10.1145/3287560.3287596[42] Fatemeh Nargesian、Erkang Zhu、Renée J. Miller、Ken Q. Pu 和 Patricia C. Arocena。2019。《数据湖管理：挑战与机遇》。PVLDB 12，12（2019），1986–1989。

[43] Fatemeh Nargesian、Erkang Zhu、Ken Q Pu 和 Renée J Miller。2018。《开放数据上的表并集搜索》。PVLDB 11，7（2018），813–825。[44] Koyena Pal、David Bau 和 Renée J. Miller。2024。《模型湖》。CoRR abs/2403.02327（2024年3月）。https://doi.org/10.48550/ARXIV.2403.02327 arXiv:2403.02327[45] Koyena Pal、David Bau 和 Renée J. Miller。2025。《模型湖》。载于第28届数据库技术扩展国际会议论文集，EDBT 2025，西班牙巴塞罗那，2025年3月25日至28日，Alkis Simitsis、Bettina Kemme、Anna Queralt、Oscar Romero 和 Petar Jovanovic（编）。OpenProceedings.org，985–995。https: //doi.org/ 10 48786 2025 81 . /EDBT. . Koyena Pal、Aamod Khatiwada、[46] Roee Shraga 和 Renée J. Miller。2023。《为表并集搜索创建生成式基准》。arXiv: 2308 03883 [ ] . cs.DB Ralph [47] Peeters、Alexander Brinkmann 和 Christian Bizer。《Web Data Commons Schema.org 表格语料库》。载于 ACM Web Conference 2024 伴随论文集。1079–1082。[48] Jason Priem、Heather Piwowar 和 Richard Orr。2022。《OpenAlex：学术作品、作者、出版场所、机构和概念的完全开放索引》。arXiv 电子印本（2022），arXiv–2205。[49] Pranav Rajpurkar、Jian Zhang、Konstantin Lopyrev 和 Percy Liang。2016。《SQuAD：用于文本机器阅读理解的100,000+ 个问题》。载于 2016 年自然语言处理实证方法会议论文集。2383–2392。[50] Nils Reimers 和 Iryna Gurevych。2019。《Sentence-BERT：使用孪生 BERT 网络的句子嵌入》。载于 2019 年自然语言处理实证方法会议暨第九届自然语言处理国际联合会议论文集（EMNLP-IJCNLP）。3982–3992。[51] Anish Das Sarma、Lujun Fang、Nitin Gupta、Alon Y. Halevy、Hongrae Lee、Fei Wu、Reynold Xin 和 Cong Yu。2012。《查找相关表格》。载于 SIGMOD，K. Selçuk Candan、Yi Chen、Richard T. Snodgrass、Luis Gravano 和 Ariel Fuxman（编）。ACM, 817 828 10 1145 2213836 2213962 – 。https://doi.org/ . / . [52] Marco Valenzuela、Vu Ha 和 Oren Etzioni。《识别有意义的引文》。[53] Alex Wang、Amanpreet Singh、Julian Michael、Felix Hill、Omer Levy 和 Samuel R Bowman。[无日期]。《GLUE：自然语言理解多任务基准与分析平台》。载于国际学习表征会议。

[54] Thomas Wolf、Lysandre Debut、Victor Sanh、Julien Chaumond、Clement De- langue、Anthony Moi、Pierric Cistac、Tim Rault、Rémi Louf、Morgan Funtowicz 等。2020。Transformer：自然语言处理领域的最新进展。载于《2020年自然语言处理实证方法会议论文集：系统演示》，38–45。[55] Junyi Wu、Chen Ye、Haoshi Zhi 和 Shihao Jiang。2023。利用知识库和文本为网页表格进行列类型预测。《Mathematics》11卷3期（2023），560。[56] Peilin Yang、Hui Fang 和 Jimmy Lin。2018。Anserini：使用 Lucene 实现可复现的排序基线。《Journal of Data and Information Quality》（JDIQ）10卷4期（2018），1–20。[57] Yiben Yang、Chaitanya Malaviya、Jared Fernandez、Swabha Swayamdipta、Ronan Le Bras、Ji-Ping Wang、Chandra Bhagavatula、Yejin Choi 和 Doug Downey。2020。用于常识推理的生成式数据增强。arXiv 预印本 arXiv:2004.11546（2020）。[58] Haojie Zhang、Ge Li、Jia Li、Zhongjin Zhang、Yuqi Zhu 和 Zhi Jin。2022。通过自适应优化子网络，有效微调预训练语言模型。《神经信息处理系统进展》35（2022），21442–21454。[59] Shuo Zhang 和 Krisztian Balog。2021。使用关键词和表格查询进行语义表格检索。《ACM Trans. Web》15卷3期（2021），11:1–11:33。https://doi.org/10.1145/3441690[60] Xuanliang Zhang、Dingzirui Wang、Baoxin Wang、Longxu Dou、Xinyuan Lu、Keyan Xu、Dayong Wu、Qingfu Zhu 和 Wanxiang Che。2024。SCITAT：一个涵盖多种推理类型的科学表格与文本问答基准。arXiv 预印本 arXiv:2412.11757（2024）。[61] Erkang Zhu、Dong Deng、Fatemeh Nargesian 和 Renée J Miller。2019。Josie：用于在数据湖中查找可连接表格的重叠集相似性搜索。载于 SIGMOD，847–864。[62] Erkang Zhu、Fatemeh Nargesian、Ken Q. Pu 和 Renée J. Miller。2016。LSH Ensemble：互联网规模域搜索。《Proc. VLDB Endow.》9卷12期（2016），1185–1196。https://doi.org/10.14778/2994509.2994534[63] Xizhou Zhu、Jinguo Zhu、Hao Li、Xiaoshi Wu、Hongsheng Li、Xiaohua Wang 和 Jifeng Dai。2022。Uni-perceiver：面向通用感知的统一预训练架构，适用于零样本和少样本任务。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，16804–16815。