新颖表搜索 [技术报告]

1st BesatKassaie滑铁卢大学大卫·R·切里顿计算机科学学院加拿大
bkassaie@uwaterloo.ca

2nd Ren´eeJ.Miller滑铁卢大学大卫·R·切里顿计算机科学学院加拿大
rjmiller@uwaterloo.ca

arXiv:2603.07235v1 [cs.DB] 7 Mar 2026

摘要——**关系数据库和信息检索领域已对避免查询结果中的冗余进行了广泛研究，但其对数据湖的影响仍基本未被探索。我们通过研究如何在大规模数据湖中发现可并集表，使其能够为给定查询表贡献新信息，来弥补这一空白。我们正式定义了新颖表搜索（NTS）问题，即查找相对于给定查询表具有新颖性的表，并确定了 NTS 的任何评分函数都应满足的两项理想属性。我们提出了一种旨在最大化句法新颖性的具体评分机制，证明该机制满足所提出的属性，并表明相关优化问题是 NP 困难。为应对这一挑战，我们开发了一种基于惩罚的高效近似技术，即基于属性的新颖表搜索（ANTs）。我们提出了三种额外的 NTS 变体，以实现句法新颖性，并引入了两项句法新颖性评估指标。通过广泛的实验，我们证明，与其他方法相比，ANTs 在各项评估指标和不同基准测试中捕获句法新颖性的表现更优，同时执行时间也最短。**

索引词—**数据湖、表搜索、新颖性**

I. 引言

长期以来，研究者一直致力于开发技术，从海量数据集集合（即数据湖）中发现并检索目标信息 [1]–[12]。一些方法允许用户通过关键词表达需求，以搜索相关数据集；另一些方法则支持用户纵向（添加更多属性）或横向（添加更多元组）扩展查询表，以支持下游数据分析任务。这些方法采用各种相关性指标来评估数据集或表与查询的相关程度，通常不考虑结果的新颖性或多样性。多样性通常被理解为覆盖面或广度要求，而新颖性则意味着避免结果中出现冗余或重复 [13]。忽略这一点可能会对数据利用和决策质量产生负面影响。例如，一位医生分析数据集，以研究某种药物对特定患者群体的副作用。如果从数据湖中发现的表虽然相关，却只包含与查询表中患者特征相同的患者，那么由于忽略了该群体的其他部分，分析结果可能会产生偏差。这是因为相关性指标通常基于各种相似度指标（包括集合重叠等句法度量，以及词嵌入或表嵌入相似度等语义度量） [12]。在许多应用中，用户希望获得包含新颖信息的相关答案，并且不同答案之间具有多样性。以数据市场为例 [14], [15]：数据买家希望获得可并集表。由于每张表都有获取成本，她通常更愿意购买可并集且包含最多新信息的表；购买多张表时，她还希望得到一个多样化或各不相同的表集合，也就是说，这些表相对于查询表以及彼此而言都具有新颖性。

给定一个查询表，可联合表搜索的目标是从数据湖中识别出最适合与之合并的表 [1]。如果两个表具有语义相似的属性，且合并这两个表在语义上有意义，那么就认为它们可合并 [1]。较新的研究则认为，如果两个表具有语义相似的属性，且属性之间的关系或表上下文相似，那么它们就是可合并的 [2], [11]。一种常见方法是利用句法特征（例如观测值）以及语义表示（例如嵌入）来计算属性之间的相似度 [1], [16]。因此，与查询表最显然可合并的表就是查询表本身。然而，识别新颖的可合并表带来了新的考量：我们要寻找的表既要足够相似，能够支持并集操作，又要有足够差异，以带来新颖性。在这项工作中，我们将新颖表搜索（NTS）正式定义为一个问题：寻找相对于给定查询表具有新颖性的可合并表。

我们将 NTS 定义为一个重排序步骤（图 1 中的步骤 2），它作用于可并集表搜索系统的输出。给定一组最初与查询表可并集的候选表，NTS 会选出其中新颖性最高的表。我们为 NTS 定义了一种特定的新颖性度量，该度量基于句法新颖性，会惩罚包含重复（冗余）数据的表（同时仍保留一定的新颖性）。我们提出了多种解决 NTS 的技术，每种技术都采用不同的策略，并评估了它们捕捉句法新颖性的能力及其效率。

ANTs 专门用于最大限度地提高表的新颖性并尽量减少重复项；GMC 是一种最初为查询结果多样化而开发的方法，我们对其进行了调整，使其针对句法新颖性；ER 是一种以实体为中心的方法，通过最大化实体解析距离，旨在返回包含新颖实体的表；SemNov 则是一种通过最大化表语义距离来实现目标的方法。

下面通过一个示例来说明。

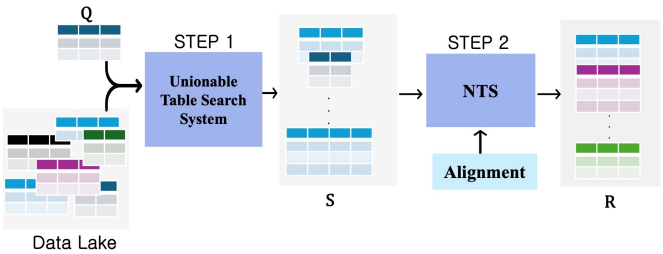


图 1：给定一个查询表 Q，NTS 将第 1 步（$|S| = k$）中的 k 可联合表作为输入，并使用新颖性感知评分函数选出 l 个最具新颖性的表（$|R| = l$）。如果搜索方法不具备内置的属性对齐功能，则会应用外部对齐器。

示例 *1.1*：图 *2(a-c)* 展示了 *Ugen-v2* 数据集 *[18]*中三个表的部分内容。*Ugen-v2* 是一个较新的表可合并性基准。用户希望通过添加更多行来扩充查询表 Q（图 *2a*），该表包含有关艺术品的信息。她将 Q 提交给一个可合并表搜索系统，并检索到一个包含 k 个候选表的排序列表。在基准测试的候选表中，可合并性最高的表与查询表有显著重叠。随后，我们的重排序组件（ANTs）将 T_1（图 *2b*）排为新颖性最高的可并联合表，并将 T_2（图 *2c*）排在第七位。ANTs（第 *IV* 节）通过兼顾较高的可合并性和较强的新颖性来识别可并联合表。请注意，T_1 与查询表的可合并性更高（共有五个属性），且更具新颖性（虽然媒介 中的值与查询表重叠，但艺术品 和艺术家中的其他值在句法上有所不同）。相比之下，T_2 的可合并性较低（仅有两个可并联合属性），且与 T_1相比，与艺术家的重叠更多。

本文的主要贡献如下：

- 我们形式化定义了识别新颖可联合表的问题，将其记为 NTS，并明确了在可联合表发现背景下，任何新颖性感知评分函数 $gscore$ 都应满足的两个一般性质。
- 我们提出了一个具体的句法评分函数，记为$nscore$，该函数满足上述性质。对于这一评分函数，我们证明了相关优化问题是 NP 困难的。
- 我们提出了一种新的基于属性的近似算法 ANTs，以高效解决 NTS。
- 我们开发了另外三种技术。其中两种方法基于句法新颖性度量：GMC（一种对现有方法的改编）和 ER（一种基于实体的技术）；此外还有 SemNov，这是一种新的语义新颖性度量，我们通过实证评估其最大化语义新颖性在多大程度上能够产生较高的句法新颖性。
- 我们证明，ANTs 还能提升下游机器学习任务的性能。

本文其余部分组织如下。第二节介绍符号记法。第三节介绍我们的评分框架，并概述实现有效新颖性评分所必需的关键属性。第四节介绍我们提出的近似算法 ANTs，用于高效识别句法上新颖的表。第五节介绍基线方法以及为进行比较而开发的各种方法。第六节讨论用于评估句法新颖性和性能的评估指标。第七节详细介绍实验设置、数据集以及对结果的全面分析。[1] 第八节介绍 ANTs 在下游机器学习任务中的应用。第九节回顾相关工作，最后，第十节总结全文并展望未来研究方向。详细证明和扩展结果见附录。

II. 预备知识

我们将数据湖表示为一个表的集合（集），记为 $\mathcal{T}$。我们将表 $T \in \mathcal{T}$ 的模式表示为 $T(A_1、\ldots\ldots、A_n)$。稍微滥用一下记号，对于表$T(A_1、\ldots\ldots、A_n)$，我们用 T（不显式列出属性）指代符合该模式的表实例，用 $A_k \in T$ 指代 T 的一个属性（列）（等价于将表 T 投影到属性 A_k 上），并用 $X \subseteq T(A_1、\ldots\ldots、A_n)$ 指代 $T(A_1、\ldots\ldots、A_n)$ 中的一个属性子集。我们用 D_i 表示属性 A_i 的域，即 $T \subseteq D_1 \times \ldots \times D_n$。我们用 $t \in T$ 表示 T 中的一个元组，记为$t = \langle v_1、\ldots\ldots、v_n \rangle$。在全文中，我们假设所有数据湖表和查询表均非空。

我们对传统关系模型进行了细微调整，允许表采用多重集（包）而非严格集合的形式，因为这在实际数据湖中很常见。除非另有明确说明，本文中的所有表上的集合运算和关系运算均在多重集的语境下解释。例如，投影算子 π 是多重集投影，因此可能产生重复元组。具体而言，当提到两个关系 T 和 T' 的并集时，我们假设其为多重集并集（包并集），记作 $T \cup T'$。在多重集中，两个元组的所有属性都可能具有相同值。为方便记法，我们可以假设每个元组都有一个唯一标识符，以便将其与其他元组区分开来。

为了对两个可进行并集运算的数据湖表（其属性名称可能不同）执行并集运算，我们使用 Nargesian 等人 [1] 定义的对齐。

定义 *1*：对齐 $\mathcal{A}(T, T')$ 是从 X 到 Y 的一一映射，其中 $X \subseteq T(A_1, \ldots\ldots, A_n)$ 且 $Y \subseteq T'(A'_1, \ldots\ldots, A'_m)$，并且 $|X| = |Y| \geq 1$。[2]

元组可能包含空值（$\perp$）。由于我们假设没有关于这些关系的其他信息，例如其创建细节、模式或可能的约束，因此不区分$\perp$ [19]的各种解释。与 SQL 不同，我们对涉及空值的比较采用简单的二值逻辑。也就是说，涉及 $\perp$ 的比较会被视为 $\perp$ 是一个特定值，因此（$\perp == \perp$ 为真，而其他任何情况 均为假）。此外，给定两个表 $T(A_1、\ldots\ldots、A_n)$ 和 $T'(A'_1、A'_m)$，以及它们对应的对齐 $\mathcal{A}(T、T')$，我们引入一个新的有序算子，记为 $\uplus(T、T'、\mathcal{A})$，它是一种非对称多重集

[1]代码和数据集可在 https://github.com/Besatkassaie/NTS/tree/master[2]在无歧义时，$\mathcal{A}(T, T')$ 缩写为 $\mathcal{A}$。

ID	艺术品	艺术家	创作日期	媒介	风格
t_1	《蒙娜丽莎》	列奥纳多·达·芬奇	1503–1506	杨木板油画	盛期文艺复兴
t_2	《干草车》	约翰·康斯特勃	1821	布面油画	浪漫主义
t_3	《奥尔南的葬礼》	古斯塔夫·库尔贝	1849–1850	布面油画	早期尼德兰绘画

（a）查询表 Q。元组从 Art-History YZMEPGTH.csv 中采样

ID	艺术品	艺术家	创作日期	媒介	风格	状况
t_1	《睡莲》	克劳德·莫奈	1897–1926	布面油画	自然	Good
t_2	《秋千》	让-奥诺雷·弗拉戈纳尔	1767	布面油画	具象	优秀
t_3	战舰无畏号	J.M.W. Turner	1839	布面油画	历史题材	Good

(b) 可并集表 T_1。元组取自 Art-History UPFR2P3Y.csv

ID	艺术品	艺术家	主题内容	文化背景
t_1	《蒙娜丽莎》	列奥纳多·达·芬奇	丽莎·盖拉尔迪尼的肖像	总体而言.....对后世肖像画的影响
t_2	《记忆的永恒》	萨尔瓦多·达利′	融化的时钟	永久馆藏.....纽约
t_3	《记忆的永恒》	萨尔瓦多·达′利	融化的时钟	现代艺术博物馆，纽约

（c）可并集表 T_2。元组取样自 Art-History CW81XE6V.csv

ID	艺术品	艺术家	创作日期	媒介	风格
t_1	《蒙娜丽莎》	列奥纳多·达·芬奇	1503–1506	杨木板油画	盛期文艺复兴
t_2	《干草车》	约翰·康斯特勃	1821	布面油画	浪漫主义
t_3	《奥尔南的葬礼》	古斯塔夫·库尔贝	1849–1850	布面油画	早期尼德兰绘画
t_4	《蒙娜丽莎》	列奥纳多·达·芬奇	⊥	⊥	⊥
t_5	《记忆的永恒》	萨尔瓦多·达利	⊥	⊥	⊥
t_6	《记忆的永恒》	萨尔瓦多·达利	⊥	⊥	⊥

（d） Q 与 T_2之间左外并的结果，即 $T' = \Cup(Q,\ T_2)$。

ID	艺术品	Artist	Date C创作d	Medium	Style	状况
t_1	《睡莲》	克劳德·莫奈	1897–1926	布面油画	自然	Good
t_2	《秋千》	让-奥诺雷´·弗拉戈纳尔	1767	布面油画	具象	优秀
t_3	战舰无畏号	J.M.W. Turner	1839	布面油画	历史题材	Good
t_4	《奥尔南的葬礼》	古斯塔夫·库尔贝	*1849–1850*	布面油画	早期尼德兰绘画	⊥

(e) 稀释的可并集表 $\tau(T_1,\ Q,\ \Delta)$ 其中 $\Delta = 0.2$。来自 Q 的元组为斜体。图 2：示例说明。每个元组都添加了一个唯一标识符，以便与其他元组区分。

扩展，用于扩展 Codd 提出的经典外并 [20]。[3]我们将此操作称为左外并，它计算两个关系 T 和 T'中元组的并集；这两个关系仅部分对齐，但会保留 T 的所有属性，即使它们在 T' 中没有对齐属性。 T'中未对齐的属性会被移除。设 X 为 T 中对齐属性的集合，$\mathcal{A}(X)$ 为 T'中的对齐属性。设 X' 为 T 中剩余的（未对齐）属性。我们将 T' 投影到对齐属性 $\mathcal{A}(X)$ 上，并向 T' 添加包含⊥的额外属性，将这个新关系称为 T''。然后， $T \cup T'$ 就是 T 和 T'' 的多重集并集。与 SQL 不同，即使所有属性都对齐，我们也采用多重集语义，不消除重复项。

示例 *2.1*：考虑图 *2* 中的表 Q 和 T_2。假设属性对齐是基于属性名称匹配的， $\cup$（Q， T_2）的结果如图 *2d* 所示。请注意， Q中的所有属性都会保留，并且该运算符使用多重集语义。对于 Q 中存在、但 T_2中不存在的属性，例如风格，源自 T_2 的元组对应条目会用 ⊥填充。

[3]$\mathcal{A}$ 可以省略以求简洁； $T \cup T'$、 $\Cup(T,\ T',\ \mathcal{A})$ 和 $\Cup(T,\ T')$ 表示相同的操作。

III. 新颖表搜索

A. NTS问题定义

给定查询表 Q，可并集性搜索问题旨在从数据湖 [1]，[11]中找出与其最可联合的 k 个表。我们采用 Fan 等人 [11] 对可并集性搜索问题的定义。也就是说，对于两个表T和 T'，可并集性评分函数 $U(T,\ T')$ 将输入表映射为范围在 $[0, 1]$内的分数。请注意，可并集性评分函数假定已计算出 T 与 T'之间的对齐。

定义 *2*：（**表联合搜索**）给定数据湖表集合 $\mathcal{T}$、查询表 Q，以及作用于表对的可并集性评分函数 U，表联合搜索确定一个子集 $S \subseteq \mathcal{T}$，其中 $|S| = k$ 且 $\forall T \in S$ 且 $T' \in \mathcal{T} - S$，我们有 $U(Q, T') \leq U(Q, T)$。

相比之下，新颖表搜索（NTS）旨在从数据湖中识别出l 最新颖且可并集的表。在这项工作中，我们将 NTS 定义为第二步（图 1 中的步骤 2），该步骤针对可并集表搜索系统的输出进行操作。具体而言，我们假设给定查询表 Q后，表联合搜索步骤已识别出一个可并集表集合 S，其中 $|S| = k$。随后，第二步根据相对于 Q的新颖性，识别出一个$R \subseteq S$，其中 $|R| = l \leq k$。为此，我们假设存在一个新颖性评分函数，其输入包括

查询表 Q 和表集合 R，并输出一个分数，用于表示 R 相对于 Q的新颖性。

定义 *3*：（**新颖表搜索**）给定查询表 Q、可联合表集合 S（其中 $|S| = k$）以及新颖性评分函数 gscore，新颖表搜索（NTS）找到子集 $R \subseteq S$，其中 $|R| = l$ 且 $\forall S' \subseteq S$，其中 $|S'| = l$ 且$R \neq S'$，我们有 gscore(Q, S') ≤gscore(Q, R)。

B. 新颖性评分函数的性质

给定查询表 Q，我们希望识别出 l 个可联合表（来自可联合表集合 S），这些表与 Q展现最大新颖性。我们确定了新颖性评分函数应满足的两个性质。

公理*1*：（明显重复公理）

$$S' = S \cup \{Q\} \wedge Q \notin S \Rightarrow \text{gscore}(S') < \text{gscore}(S)$$

该公理指出，如果结果集中包含查询表，则该搜索结果的新颖度分数必须低于不含 Q的集合。

给定查询表 Q 以及可并集表 T，T 中任何已存在于 Q的元组都不应为搜索结果的整体新颖性做出贡献，且与 T 中存在的其他元组无关。为正式定义这一点，我们首先定义 T 的稀释版本 。

定义 *4*：给定查询 $Q \in Q(B_1, \ldots\ldots, B_z)$、可并集表 $T \in T(A_1, \ldots\ldots, A_n)$、它们对应的对齐$\mathcal{A}$以及稀释程度 $0 < \Delta \leq 1$，相对于 Q 的 T 的稀释版本记为 $\tau(T, Q, \Delta)$，其中包含与 $\lceil \Delta \cdot |Q| \rceil$ 个来自 Q 的元组作外并集的 T。

$$\tau(T, Q, \Delta) = \uplus(T, Q', \mathcal{A})$$

其中 $Q' \subseteq Q$, $|Q'| = \lceil \Delta \cdot |Q| \rceil$。

示例 *3.1*：假设图 *2b* 中的表 T_1 仅包含所示元组，则相对于 Q 的 T_1 的一种可能稀释，记为 $\tau(T_1, Q, \Delta)$，对于 $\Delta = 0.2$ 而言，如图*2e*所示。

公理 *2*：（稀释公理）

$$S' = S \cup \{\tau(T_m, Q, \Delta)\} \wedge T_m \in S \Rightarrow \text{gscore}(S') < \text{gscore}(S)$$

请注意，这两条公理都是句法的，因为它们指出新颖度分数应对包含元组完全副本的情况予以惩罚。它们并未涉及语义相似的元组。接下来，我们将介绍一个具体的句法评分函数 $nscore$，作为 $gscore$ 的一个具体实例，并证明它满足这些公理。

C. 新颖性评分

为了计算表的新颖度分数，我们使用元组新颖度分数。给定一对元组 $t = \langle v_1, \cdots v_n \rangle$ 和$t' = \langle v'_1, \cdots, v'_n \rangle$ 我们期望元组新颖度分数满足以下条件：**C1**）当两个元组相同时，元组新颖度分数最小；**C2**）当两个元组都包含全部非空值，且没有对应的属性值相同时，元组新颖度分数最大。基于这一思路，我们定义元组新颖度分数。

定义*5*：（**元组对新颖度分数**）给定两个具有相同元数的元组 n、$t = \langle v_1, \cdots v_n \rangle$ 和 $t' = \langle v'_1, \cdots, v'_n \rangle$，它们的元组对新颖度分数定义如下：

$$\text{tuple_nscore}(t, t') = \frac{1}{n} \sum_{i=1}^{n} \textit{novelty}(v_i, v'_i) \quad (1)$$

s.t.

$$\textit{novelty}(v_i, v'_i) = \begin{cases} 1 & v_i \neq v'_i \wedge v_i, v'_i \neq \perp, \\ \beta_i & v_i \neq v'_i \wedge (v_i = \perp \vee v'_i = \perp), \\ 0 & \textit{else}. \end{cases}$$

其中 $0 \leq \beta_i \leq 1$。注意，$0 \leq$元组 nscore$(t, t') \leq 1$。

对于每一对不同的常量，我们赋予新颖度 1。当一个属性值为空值时，我们赋予不同的奖励β_i，其依据是空值可能与给定的常量相等。这种情况发生的可能性取决于属性 i的域中重复值的数量。令P_i 表示两个随机元组在属性 i 上共享相同（非空）值的概率，则我们设定 $\beta_i = 1 - P_i$。最后，当两个值都为空值时，新颖度为零。

定义*6*：**(元组新颖度)**我们将表 T 中元组 t 的新颖度定义为最小元组 nscore：

$$N(t) = min\{\text{tuple_nscore}(t, t') : t' \in T \wedge t \neq t'\}$$

示例*3.2*：对于图*2e*中的表，t_3 的元组新颖度是元组对 $\{(t_3, t_1)$、(t_3, t_2)、$(t_3, t_4)\}$ 的新颖度中的最小值。属性 *Condition* 包含一个空值和两个相等的值，因此其对应的 β 根据数据计算得出为 $1 - \frac{1}{3} = \frac{2}{3}$。对于 *Medium*，只有一个值，因此 $P_{Medium} = 1$，所以其 β 为 0，这意味着该属性中的空值不会带来新颖性。此示例中的所有其他属性都只有唯一值，这意味着 $\beta = 1$。然而，由于唯一的空值位于 *Condition* 中，对于这个简单的示例，我们只需要 $\beta_{Condition}$。因此，

$$N(t_3) = \min\{\frac{4}{6}, \frac{5}{6}, \frac{4 + 1 \times \frac{2}{3}}{6}\} = \frac{4}{6}$$

直观地说，与 t_1 相比，t_3 提供的新颖性最低（因为它们的重叠最多）；与 t_2 相比，则提供的新颖性最高。参数 β 则对空值的新颖性进行建模，使其介于相同值（t_1）和不同值（t_2）所对应的新颖性之间。

定义*7*：**(表新颖度分数)**给定一个包含y 个元组 $|T| = y$ 的表 T，T的新颖度分数定义为：

$$\text{table_nscore}(T) = \frac{1}{y} \sum_{t \in T} N(t) \quad (2)$$

因此，表的新颖度分数定义为表中所有元组的新颖性之和除以元组总数。我们会仔细考虑元组数量以及每个元组带来的新颖性程度。要获得较高的新颖度，我们期望每个元组都能在新颖性方面带来有意义的增值，即 $N(t) > 0$。在每个表（因此也包括每个元组）都有获取成本的数据市场环境中，新颖度分数衡量单位成本的新颖性，并对包含许多冗余元组（即具有 $N(t) = 0$）的表予以惩罚。

要计算查询表 Q 相对于 l 个可并集表的新颖度分数，我们将查询表与这 l 个可并集表进行左外并集，并计算所得结果的表新颖性。

定义 *8*：**（搜索新颖性得分）**给定一个查询表 Q、一组 l 个可并集表 $R = \{T_1, \cdots T_l\}$，以及从 Q 到每个 T_i 的一组对齐关系，记为 $\mathbb{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_l\}$；搜索的新颖度分数 nscore($Q$, R, $\mathcal{A}$ 定义如下：

$$\text{nscore}(Q, R, \mathbb{A}) = \text{table_nscore}(\breve{T})$$

其中 T 是：

$$\breve{T} = Q \cup \bigcup_{T_i \in R} \Cup(Q_\emptyset, T_i, \mathcal{A}_i) \tag{3}$$

其中，$Q_\emptyset = Q \setminus Q$ 表示一个保留与 Q相同模式的空表。□

通过将 Q纳入搜索新颖性得分（nscore）的计算，我们将查询表上的新颖性反映在结果的整体新颖度分数中。接下来，我们证明搜索新颖性得分函数满足前文提出的公理。

定理 *1*：搜索新颖性得分（定义 8）满足显而易见的重复项公理和稀释公理。□

我们将新颖表搜索（NTS）问题定义为：从 k 可联合表中选择 l 个表，使搜索结果的搜索新颖性得分最大化。

定义 *9*：**（新颖表搜索）**给定一个查询表 Q、一个由 k 个可联合表组成的集合 $S = \{T_1, \cdots T_k\}$，以及一个从 Q 到 S 的对齐集合，记为 $\mathbb{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_k\}$，NTS 会找到一个由 l 个表组成的集合 R，其中：

$$R = \underset{S' \subseteq S, |S'|=l}{\text{argmax}} \ \text{nscore}(Q, S', \mathbb{A}) \tag{4}$$

为了找到 NTS 的精确解，我们必须评估来自 S 的所有大小为 l 的子集。对于每个子集，我们计算其新颖度分数，从而导致数量呈指数级增长的评估。结果是，找到精确解在计算上不可行，尤其是对于较大的 k 和 l。

定理 *2*：为新颖表搜索问题找到最优解是 NP 困难。□

为了解决这一复杂性，我们将在下一节提出一种近似算法。

IV. 基于属性的新颖表搜索

为了近似 nscore，我们采用一种基于属性的方法，通过估计属性的新颖性，而非逐一遍历元组。我们的解决方案（ANTs）估计对齐属性之间的句法不相似度（新颖性）分数。ANTs 会返回一个有序的表列表，因此可用于对表联合搜索系统生成的表列表进行排序（或重新排序）。在这一计算中，我们尽可能提高属性的句法不相似度，同时仍优先考虑可并联合（语义相似）的属性。我们的方法以两个函数为基础，分别计算每对对齐属性的句法相似度和语义相似度。如果两个属性的值域包含相同或密切相关的概念，那么它们在语义上相似。由于我们希望优先考虑更具可联合性的属性，因此遵循可联合性研究中的做法，将每个属性表示为语义向量空间中的一个嵌入，并以它们嵌入之间的距离作为语义相似度的度量。

A. 属性的句法相似度

抽象地说，我们需要一个函数 syn sim(B_i, A_j) $\in [0, 1]$，用于表示两个属性$\langle B_i, A_j \rangle$的**句法相似度**。如果两个属性的值域包含许多相同值，那么它们在句法上相似。我们通过各属性的观测值域来刻画属性，并以这些值的重叠情况或分布作为句法相似度的度量。请注意，通常会使用集合相似度度量（例如杰卡德）来衡量句法相似度。这种方法适用于大值域。然而，对于小值域，简单的集合重叠可能并不足够。例如，考虑两个包含一周中各天的属性（最多七个值）。一个可能表示在线销售，其值分布涵盖一周中的所有七天；另一个可能表示乐队巡演日期，也包含全部七天，但明显偏向周末。两者的杰卡德相似度为 1，但其分布相当不同。因此，对于小值域，我们改为利用值的分布，并将它们之间的差异作为句法新颖性的信号。

我们定义了一个名为 s 的超参数，用于区分某个属性属于小域还是大域。该值根据数据湖的特征通过经验确定，因为它会影响排序算法的有效性。第七节-E 讨论了我们对该参数的研究。

大值域的句法相似度 $\mathcal{V}(B_i)$ 和$\mathcal{V}(A_j)$分别是属性 B_i 和 A_j 的值域。如果这些值域的并集大小大于 s，我们就使用杰卡德相似度 [21] 来计算属性的句法相似度。

小值域的句法相似度对于小值域属性，我们计算每个属性中已观察到的（非 $\perp$）值的分布，并使用这些分布之间的距离作为新颖性的指标。我们分别用$\mathbb{B}_i$ 和 $\mathbb{A}_j$表示与属性 B_i 和 A_j 相关联的离散概率分布。这些分布是在值域的并集上计算的，因此每个分布都可能包含一些缺失值。对于缺失值，我们赋予概率 0。

接下来，我们需要一种度量两个概率分布之间差异的指标，即散度度量，并且该指标需满足以下性质：i) 取值范围有界；ii) 存在缺失值时仍有定义；iii) 具有对称性，因为不存在参考分布，且两个属性（B_i 和 A_j）受到同等对待。詹森-香农散度（JSD）具备这些理想性质，与我们的问题 [22]–[24].[4]相契合。

示例 *4.1*：假设图 *2(a–b)* 中的表仅包含所示元组。考虑图 *2* 中的查询表 Q，以及可并集表 T_1（图 *2b*）和 T_2（图 *2c*）；在这两种情况下，属性艺术家都与 Q.艺术家对齐。直观来看，Q.艺术家与T_1.艺术家之间的 *JSD*（JSD $= 1$）大于 Q.艺术家与 T_2.艺术家之间的 *JSD*（JSD $= 0.82$），因为在前一对中，

[4]有关 Jensen–Shannon 散度的更多详细信息，请参阅附录 B。

经验分布的概率质量都集中在不同的 艺术家身上，而后一对中的两个分布都将相同的概率质量分配给列奥纳多·达·芬奇。这部分共享的概率质量使两个分布更加接近，从而得到较小的 *JSD*。

对于小值域，我们将 syn sim(B_i, A_j) 定义为 1 减去它们概率分布的 Jensen–Shannon散度。

定义*10*：**(属性句法相似度)**对于一对对齐属性 $\langle B_i, A_j\rangle \in \mathcal{A}(Q,T)$，令 $\mathcal{V}(B_i)$ 和$\mathcal{V}(A_j)$ 为它们的值域，且 $\mathcal{D} = |\mathcal{V}(B_i) \cup \mathcal{V}(A_j)|$。此外，令 $\mathbb{B}_i$ 和 $\mathbb{A}_j$ 表示它们在 $\mathcal{V}\ B_i\ \cup \mathcal{V}\ A_j$ 上的离散概率分布。

$$\text{syn_sim}(B_i, A_j) = \begin{cases} \frac{|\mathcal{V}(B_i)\cap\mathcal{V}(A_j)|}{|\mathcal{V}(B_i)\cup\mathcal{V}(A_j)|} & \mathcal{D} > s \\ 1 - \text{JSD}(\mathbb{B}_i, \mathbb{A}_j) & \mathcal{D} \le s \end{cases} \quad (5)$$

B. 属性的语义相似度

为了对一对对齐属性之间的语义相似度进行建模，我们按照 Fan 等人 [11], 提出的方法，计算其语义表示的余弦相似度；该方法在可合并表发现方面取得了最先进的性能。对于属性 A，令 Sem(A) 表示 Starmie 学得的属性嵌入。

定义*11*：**(属性语义相似度)** 对于一对对齐属性 $\langle B_i, A_j\rangle \in \mathcal{A}(Q,T)$，$B_i \in Q$、$A_j \in T$，语义相似度定义如下：

$$\text{sem_sim}(B_i, A_j) = \text{cosSim}(\text{Sem}(B_i), \text{Sem}(A_j)) \quad (6)$$

C. 新颖性感知可并表性

现在，我们将属性新颖性 定义为一种分数：该分数最大化时，句法相似度最小化，语义相似度则最大化。

定义 *12*：**（属性新颖性）**对于一对对齐属性 $\langle B_i, A_j\rangle \in \mathcal{A}(Q, T)$，$B_i \in Q$、$A_j \in T$，属性新颖性AttNovelty($B_i$, A_j) 定义如下：

$$(1-\text{syn_sim}(B_i, A_j))^b \times \text{sem_sim}(B_i, A_j) \quad (7)$$

其中，$b \ge 0$ 是一个超参数。 □

属性新颖性旨在同时捕捉属性对的可并表性和新颖性。对于可并表性高且新颖性高的属性对，该分数达到最大；当两者都较低时，该分数达到最小。超参数 b 使我们能够调整这两种信号的相对重要性。当 $b = 1$时，句法信号和语义信号的权重相等。较小的 b会使语义相似度的权重更高，而较大的b 则会使句法不相似度（新颖性）的权重更高。

因此，表的新颖性是与查询表对齐的属性的新颖性的总和。

定义 *13*：**（表新颖性）**给定查询表 Q 和可并集表 T，表新颖性函数定义为对其对齐属性的属性新颖性进行聚合：

$$\text{TableNovelty}(Q,T) = \sum_{\langle B_i, A_j\rangle \in \mathcal{A}(Q,T)} \text{AttNovelty}(B_i, A_j) \quad (8)$$

直观而言，我们更倾向于选择与查询表在语义上相似、且高度可并集，同时包含句法上不同（新颖）值的表。因此，我们会惩罚语义相似度，并依据对齐属性的句法相似度进行调整。

示例 *4.2*：接着示例 *4.1*，令 $b=1$和$s=5$，并假设所有对齐属性的语义相似度1。T_1 的每个属性都有一个与之对齐的对应属性。除 *Medium* 外，这些对齐属性对的值域都很大且互不相交，因此其属性新颖度 最大1。对于 *Medium*，两种分布在 *Oil on canvas* 上重叠，因此 *JSD* 为0.44。因此

$$\text{TableNovelty}(Q, T_1) = 1+1+1+0.44+1 = 4.44$$

相比之下，T_2 只有两个属性与 Q对齐。T_2 *Artwork* 在句法上具有新颖性，与 Q *Artwork* 在语义上相似，其属性新颖度 最大1。T_2 *Artist* 与 Q *Artist* 共享概率质量，*JSD* 为0.82，因此句法新颖性较低：

$$\text{TableNovelty}(Q, T_2) = 1+0.82 = 1.82$$

D. ANTs 解决方案

总而言之，ANTs 的输入包括查询表 Q、由 k 个S 可联合表组成的集合、Q 与每个表 $T \in S$之间的属性对齐，以及要返回的表数量 l。它会穷举扫描所有可联合表 $T \in S$，使用 TableNovelty(Q, T)计算其新颖度分数，按新颖度降序排列候选项，并返回最具新颖性的前 l 个可联合表（算法 1）。

算法 1： ANTs

输入： Q: 查询表； S: 由 k 个可联合表组成的集合； A: 属性对齐； l: 要返回的表数量

输出： R: 最具新颖性的可并集表列表

```
分数 ← {} * 表到分数的映射
对每个 T ∈ S 执行
    A ← A[T] * 获取对齐
    分数[T] ← 获取表新颖度(Q, T, A) * 定义 13
分数 ←按值排序(分数)* 降序排序
R ← top(scores, l) * 按新颖度排名
返回 R
```

V. 基线方法与对比技术

在评估中，我们考察了五种技术：一种基线方法、三种 NTS 变体，以及一种最大–最小多样化启发式，它作为近似 nscore 基线。下面，我们总结各项技术的主要特点。

Starmie： Starmie [11]是一款用于数据湖中数据集发现的先进集成框架，可用于表联合搜索。它采用对比学习方法，以完全无监督的方式训练属性编码器。这些属性编码器能够捕捉表中的上下文语义。Starmie通过计算属性嵌入向量之间的余弦相似度，得出属性的可并集性得分，并根据对齐属性汇总这些得分，得出表的总体可并集性得分。

Starmie采用其内置的对齐机制，该机制基于最大二分图匹配。我们将Starmie用作基线，根据可并集性得分相对于给定查询对排名前 k 的可联合表进行排序。第七节介绍了Starmie所用的超参数设置详情。

边际贡献贪心算法（GMC）：虽然我们的工作首次将多样性（新颖性）引入表搜索，但查询结果中的多样性已在先前的研究中得到探讨 [17]。Viera等人 [17] 比较了多种查询结果多样化方法，并表明一种称为GMC的方法在结果质量（精确率和最优性）与执行时间之间实现了最佳权衡。我们希望研究将该方法调整为用于解决NTS时的表现。给定一个查询（可能是文档、图像，或在我们的案例中是表），Vieira等人 [17]旨在检索一组与查询高度相似的 l 元素，同时确保所选集合彼此具有多样性，从而形成一个*k*相似多样化集合。令 $S' \subseteq S$，其中 S 是候选结果集合。令 $\delta_{\text{sim}}(q,\ S')$表示查询 q 与 S'中元素之间的相似度，并令 $\delta_{\text{div}}(S')$衡量 S'中元素之间的多样性。给定参数 $\lambda \in [0,1]$，Vieira等人 [17] 将目标函数表述为一个最大和优化问题，以确定 R:

$$R = \arg \max_{S' \subseteq S, l=|S'|} \mathcal{F}(q, S') \tag{9}$$

s.t.

$$\mathcal{F}(q,S') = (l-1)\times(1-\lambda)\times\delta_{sim}(q,S') + 2\times\lambda\times\delta_{div}(S')$$

由于寻找最优集合 R 是 NP 困难，Vieira 等人 [17] 提出了 GMC，这是一种用于识别 R的确定性近似算法。GMC 与所有已知的、仅依赖相关性和多样性分数来实现查询结果多样化的方法进行了比较；结果表明，在取得更高的目标函数值方面，GMC 始终优于其他方法$\mathcal{F}$（$q,\ S'$）。因此，我们仅将本方法与 GMC 进行比较。接下来，我们将介绍如何针对我们的设定调整 GMC。

给定查询表 Q和一组可联合表 S，我们使用 GMC 算法选出 l 个可联合性最高且最新颖的表。为此，我们向 GMC 提供两个函数：**1**）一个用于衡量查询表 Q 与数据湖表 $T \in S$ 之间相似度（即可联合性）的函数，记为 δ_{sim}（$Q,\ T$）：$Q \times S \to \mathbb{R}^+$，我们使用 sem sim（定义**11**）；**2**）一个用于量化两个数据湖表 $T,\ T' \in S$之间句法新颖性的函数，记为δ_{div}（$T,\ T'$）：$S \times S \to \mathbb{R}^+$，我们使用 $1-$syn sim（定义**10**）。按照 Vieira 等人 [17],的建议，我们在所有实验中均设置 $\lambda = 0.7$。

GMM：我们还使用了一种名为 GMM 的近似算法，它是一种针对最大最小多样化问题、具有理论近似保证的贪心算法。最大最小多样化问题旨在选择一个项目子集，使其中的最小成对距离最大化 [25]。形式化地，给定一个由 k 个元素组成的全集 $\mathcal{U}$ 以及一个大小为 l 的子集，目标是找到一个大小为 l 的子集 S，使 $div(S)$ 最大化：

$$S = \arg \max_{\substack{S' \subseteq \mathcal{U} \\ |S'|=l}} div(S')$$

其中，$div(S) = \min_{s,\ s' \in S,\ s \neq s'} \delta(s,\ s')$ 用于衡量集合 S 中元素之间的最小成对距离。Ravi 等人 [25] 证明，GMM 对最大最小多样化目标函数可达到$_1$ $_2$近似保证，即其所达到的 $div(S)$ 经证明至少为最优值（可能达到的最佳值）的 50%。因此，GMM 提供了一个近似 nscore 基线，仅 关注句法新颖性（多样性）。我们采用 Moumoulidou 等人 [26], 提出的 GMM 变体，该变体允许使用给定元素进行初始化。我们使用先前定义的句法距离函数 δ_{div}，衡量表之间的成对句法距离（见第 IV 节）。

句法新颖性的语义表示（SemNov）：接下来，作为另一项基线，我们研究使用表嵌入之间的距离作为语义新颖性信号是否能够产生句法新颖性，并将其有效性与 ANTs 进行比较。TABBIE 等表嵌入 [27] 捕获了表的底层语义，而表嵌入之间的距离可作为语义差异的代理，进而作为表之间语义新颖性的代理。我们提出，使用两个属性对应的表嵌入之间的语义相似度，对这两个属性之间的语义相似度进行惩罚。请注意，在此情况下，我们不区分不同的域大小。因此，对于任意一对对齐属性 $\langle B_i,\ A_j \rangle \in A(Q,\ T)$，其中$B_i \in Q,\ A_j \in T$，AttrSemNovelty($B_i,\ A_j$) 将为：

$$\begin{aligned}\text{AttrSemNovelty}(B_i, A_j) = (1 - \text{table_sim}(Q,T))^b \\ \times\ \text{sem_sim}(B_i, A_j)\end{aligned} \tag{10}$$

其中，table sim($Q,\ T$) 表示从经 TABBIE 训练的模型中提取的、代表表 Q 和 T 的向量之间的余弦相似度，其取值范围为 $[0,1]$。同样，此处的表新颖性就是语义 AttrSemNovelty 分数之和。

新颖表搜索的实体解析（*ER*）：ANTs 采用基于属性的策略，以降低新型表搜索问题的时间复杂度。另一种做法是采用基于元组的近似方法。我们应用实体解析技术来估算数据湖中查询表与候选表之间的重叠度，即匹配实体的数量，然后对表进行排序，使重叠（近似元组重叠）较少的表获得更高分数。实体解析过程包含两个主要步骤：阻塞和匹配。查询表到达后，首先将其元组与每个可并集表的元组进行阻塞，随后仅在共享块内执行匹配。由于我们的工作基于句法新颖性，因此在实体解析任务中采用句法匹配方法 [28]。在阻塞方面，我们试验了文献中的若干技术，例如 ScaNN [29], kNN –Join [30], 和 DeepBlocker [31]，并采用了速度最快的方法 ScaNN。附录 D 对我们选择这些阻塞技术的原因作了详细说明。在实验中，我们将该系统称为 ER。

VI. 评估指标

我们将使用搜索新颖性得分（nscore）（定义 8）来评估 NTS 解决方案的质量。不过，nscore 的计算开销很大，因此我们只对小型基准测试报告该指标，并采用若干计算成本较低的指标。查询表与结果表取并集后，搜索新颖性得分会降低（公式（3））。因此，直观而言，如果数据湖中存在查询表的精确副本，那么将该副本排在前列的技术就不太理想。我们使用明显重复项指标来衡量这一倾向：对于每个查询，该指标判断查询表的精确副本是否出现在前-l 个结果中。同样，如果前-l 个结果中的任何可并集表包含查询表的元组（稀释），也会损害搜索新颖性得分。根据经验，我们更倾向于优先排列未被稀释的可并集表，并使用句法新颖性度量（SNM）量化这一行为。

A. 明显重复项

公理1中的“明显重复项”一词指的是与查询表重复的可并集表。受该公理启发，我们提出了一项评估指标，用于评估系统是否会将查询表的此类明显重复项排在前 l 个结果之中，前提是数据湖中存在这些重复项。给定一个查询 Q和一个表集合 R，由一种新颖表搜索方法返回，其中 $|R|=l$，我们将该指标定义如下。

$$\textit{blatant-duplicate}(Q,R)=\begin{cases}1, & \text{if } Q\in R,\\ 0, & \text{else.}\end{cases} \tag{11}$$

为了在实验中使用该指标，我们将查询表自身纳入针对NTS所考虑的 k 可并集表集合中。

B. 衡量句法新颖性

基于二元偏好（bpref） [32],这一信息检索领域中已确立的评估指标，我们提出了两项新指标，用于评估搜索算法从数据湖中识别新颖表的有效性。二元偏好旨在用于相关性评估不完整的评估场景。bpref指标会在一个已判定的不相关文档排在一个已判定的相关文档之前时对系统进行惩罚，但不会受到检索到未判定文档的影响。

类似地，对于查询 Q 、一组 k 可联合表 $S=\{T_1,\cdots,T_k\}$及其对应的对齐 $\mathcal{A}$，以及稀释度 Δ，我们构造一个新集合 $\mathcal{S}=S\cup S_{dil}$，其中 $S_{dil}=\{\tau(T_i)|T_i\in S\}$。随后，我们使用一种称为句法新颖性度量（SNM）的度量，评估搜索算法在对 $\mathcal{S}$ 中的元素排序并选取排名靠前的 l 个表时，能否有效地将每个可联合表 T_i 排在其对应项 $\tau(T_i)$ 之前。

定义 *14*：（**句法新颖性度量（SNM）**）设S 是由 k 个可联合表组成的、针对查询 Q的集合，并且 $S_{dil}=\{\tau(T_i)|T_i\in S\}$是稀释度为 Δ的稀释表集合。假设新颖性搜索算法返回一个大小为 l的有序列表 $S'\subset S\cup S_{dil}$。

$$\begin{aligned}\textit{SNM}&=1-\frac{|O|+|Y|}{l} \qquad (12)\\ \text{s.t.}\quad O&=\{T_i\mid(\tau(T_i)\in S'\wedge T_i\notin S')\\ &\qquad\vee(T_i=Q\wedge\tau(T_i)\notin S'\wedge T_i\in S')\},\\ Y&=\{T_i\mid(\mathcal{I}(S',T_i)>\mathcal{I}(S',\tau(T_i)))\\ &\qquad\vee(T_i=Q\wedge\mathcal{I}(S',T_i)<\mathcal{I}(S',\tau(T_i)))\}.\end{aligned}$$

$\mathcal{I}(S',\ T)$表示表 T在有序列表S'中的位置。索引越低，表明该表的排名越好。直观而言， O 表示出现在前 l 个结果中、但其对应的原始表并未一同出现的稀释表集合；而 Y表示在前 l 项中，稀释版本的排名低于（即被视为优于）其对应原始表的表对数量。为避免奖励返回明显重复项的系统，我们在 O 和 Y的公式中对检索到明显重复项的情况施加惩罚。请注意，$0\leq$SNM ≤ 1。对于给定查询，当仅返回可联合表的稀释版本时，SNM 的最小值为 0。

SNM 适用于基于排序的 NTS 方法。由于我们的 GMC 方法返回的是一个集合，而不是排序列表，我们还定义了一种无排序变体。

定义 *15*：（**简化 SNM（SSNM）**）简化 SNM 定义如下：

$$\textit{SSNM}=1-\frac{|O|}{l} \tag{13}$$

C. 附加指标

鉴于 GMC（第 V 节）的计算开销较大，我们旨在研究 ANTs 在多大程度上能够优化为 GMC 定义的目标函数。作为一项附加评估指标，我们根据系统的 $\mathcal{F}(Q,\ S')$ 进行比较，或者简单地根据 $\mathcal{F}$ （公式（9））进行比较。

VII. 实验

我们使用 Python 3.8.5 实现了 ANTs、ER、GMM 和 GMC，而 SemNov 由于库依赖项的原因需要使用 Python 3.6。所有实验均在一台服务器上进行，该服务器配备五颗 AMD EPYC 9554 64 核处理器和 512 GB 内存。我们使用公开提供的 Starmie 代码库。[5]

A. 数据集

我们首先详细介绍三个数据集，并概述其特征以及我们如何针对实验设置对其进行调整。准备数据集时，我们主要考虑两点：i）我们将新型表搜索问题视为在可并集表搜索系统之上的一个步骤（见定义 3）。为消除初始搜索系统性能的影响，对于每个数据集中的每个查询表，我们仅考虑 k 可并集表（由真实标注确定）。随后，我们仅对这些可并集表应用各种方法，以找出排名前 l的最终表。我们设置 $k=20$；ii）评估的一个关键环节是，在可联合候选项池中系统地纳入数据湖表的稀释版本以及查询表本身（作为明显重复项）。在数据集准备过程中，我们将稀释程度设为

[5]https://github.com/megagonlabs/starmie

（Δ）0.4，适用于所有数据集。请注意，数据湖中的所有表都有一个稀释版本。还请注意，除评估第七节 E 中对齐质量影响的消融实验外，我们在所有实验中均使用真实标注对齐，且所有表均非空。

表联合搜索。TUS 数据集由 10 个基础表合成，这些基础表来自英国和加拿大的开放数据 [1]。我们对这些基础表依次应用投影和选择操作，生成可并集表。我们使用与 Khatiwada 等人所用相同的 TUS 版本 [2].[6]在此数据集中，查询共享可并集表，也就是说，单个数据湖表可能与多个不同的查询表可并集。由于稀释数据湖表是数据集准备的一部分，为简化实现过程，我们使用了一组查询子集，其中各查询之间不共享可并集的数据湖表。最终的 TUS 数据集包含 42 个查询表和 1008 个可并集数据湖表，其中包括 504 个稀释表。

Santos。借鉴 Nargesian 等人提出的类似基准创建技术 [1]., Santos Small 同样使用开放数据构建 [2]。与 TUS 类似，我们仅保留一组与其他查询不共享可并集表的查询。重新利用后的 Santos Small 数据集包含 35 个查询和 948 个数据湖表，其中包括 474 个稀释版本。我们将此数据集简称为 Santos。

Ugen-v2。最新的可联合表基准由大语言模型生成，***Ugen-v2*** [18]。与 TUS 和 Santos 等人工整理的基准相比，该基准对所有可联合表搜索解决方案都更具挑战性。我们创建了一个规模较小的 Ugen-v2 版本，包含约 40% 的 Ugen-v2 查询，这些查询的元组数量最少；此外，还包含 19 个涉及两项考量的查询。首先，为了消除自动对齐可能造成的任何负面影响，我们选取了一部分查询及其对应的可联合表，并手动对齐其属性。其次，对于中等规模的数据集，计算搜索新颖性得分（定义 8）耗时较长。为了能够计算 nscore，我们创建了这个具有挑战性的数据集的较小版本。为简洁起见，我们将这个基准的较小版本简称为 Ugen-v2。

B. 实验设置

我们详细介绍了实验对比中各系统的情况，包括其配置。

Starmie：我们将 Starmie [11] 用于两个目的：（1）在离线模式下生成上下文化属性嵌入，供所有方法后续使用；（2）在在线模式下，作为基线方法，根据查询表对可联合表进行排序。我们使用 Starmie 模型生成的嵌入来表示属性，并利用这些嵌入计算 ANTs、SemNov 和 GMC 中属性之间的语义相似度。Starmie 在 Santos 数据集 [33], 上进行了微调，随后使用所得模型生成属性嵌入，所有数据集的向量维度均为 768。[7] 我们使用以下超参数配置微调过程：批次大小 $=64$、学习率 $=5e-5$、最大。序。 列长度 $=128$、投。 影维度.=768、增强操作 $=$ 删除列、采样方法 =tfidf 实体、表顺序 $=$ 列，以及训练轮数 $=3$。超参数详情请参阅 Starmie 官方代码库。在在线模式下，如果只评估真实可并表，可扩展性并非主要问题。因此，为了对排名前列的 l 个可联合表进行排序，我们采用一种简单的线性扫描方法，并设置相似度阈值 τ=0.7（参见 Fan 等人所著的“在线查询处理”一节 [11]）。TABBIE：我们在推理模式下使用其公开代码库中的预训练 TABBIE 模型 [27] 。[8] 它用 768 维嵌入表示每张表。请注意，TABBIE 用于为 SemNov 创建语义嵌入。

ER：实验使用 ScaNN Python 软件包（v1.4.2）的默认设置，并将邻居数设为 $=5$，该数值是通过经验确定的，以实现快速阻塞。[9] 对于实体匹配，我们使用公开可用的 *py-entitymatching*（版本 0.4.2）代码库中提供的基于规则的技术。[10] 我们比较对齐属性的值。如果莱文斯坦相似度为 ≥ 0.85，则认为这些值相似。

C. 有效性与可扩展性的结果

ANTs 返回新颖可合并表的排序列表。为评估排序质量，我们要求至少有两个结果，并在 l 范围内改变 $[2,10]$。请注意，在 Ugen-v2 数据集中，79%（15/19）的查询所对应的可联合表少于 12 个，其中两个查询的明显重复项分别是查询本身及其稀释版本；我们在结果中对此进行了考虑。

明显重复度量。像 ANTs 和 SemNov 那样通过惩罚引入新颖性，在 Santos 数据集上能取得更好的结果（表 I）；而在 TUS 上，ANTs、SemNov 和 ER 的表现最佳（表 I）。在 Ugen-v2 数据集上，依靠句法相似度进行匹配的 ER 实现了最低重复率。

表 I：每个数据集中，在前 l 个结果（$l \in [2,10]$）内出现明显重复项的查询平均百分比。数值越低，性能越好。在所有表中，**粗体** 数字表示每行的最佳性能；下划线标出的数字为次佳。

	Starmie	GMC	ER	SemNov	ANTs
Santos	99.4%	81.9%	8.6%	**0%**	**0%**
TUS	100%	99.5%	**0%**	**0%**	**0%**
Ugen-v2	100%	98.8%	**43.6%**	52.6%	43.9%
Ugen12	100%	100%	**0%**	**0%**	**0%**
Ugen $l=2$	100%	89.5%	**0%**	**0%**	**0%**

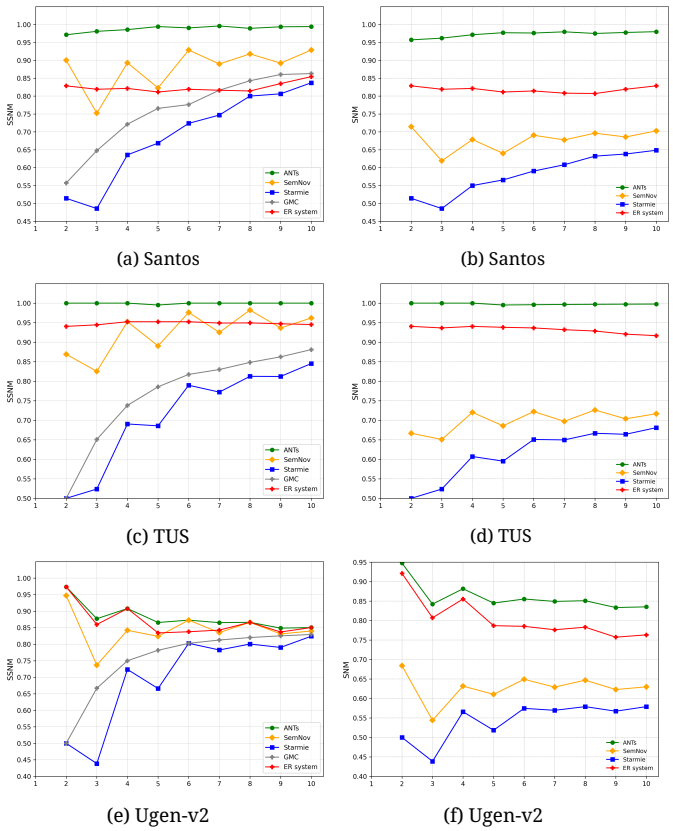
如前所述，Ugen-v2 数据集在 79% 的查询中包含数量有限的可联合表。为实现更公平的比较，我们将评估范围限定为至少包含 12 个可联合表的查询（*Ugen12*）。在这种

[6]https://zenodo.org/records/14151800

[7]Starmie 在 Santos 上进行了任意微调，因为预计它无需进一步调优即可泛化到各个数据集。由于所有方法都依赖于同一个微调后的模型，因此比较是公平的。[8]https://github.com/SFIG611/tabbie [9] https://pypi.org/project/scann/, [10] https://pypi.org/project/py-entitymatching/

状况下，ANTs、SemNov 和 ER 的行为完全相同。由于所有查询都至少包含四个可联合表，我们还报告了 $l = 2$ (*Ugen* $l = 2$) 的结果。在此设置下，ANTs、SemNov 和 ER 再次表现出相同的行为，并优于其他方法。综合考虑这些状况以及 ER 在 Santos 上的性能，ANTs 和 SemNov 总体上仍是表现最佳的方法。

句法新颖性度量。SNM 指标评估系统将原始表排在其稀释版本（新颖性较低）之前的能力，且仅适用于基于排序的方法；因此，GMC 不在评估范围内。图 3b、3d 和 3f 报告了各数据集所有查询的平均 SNM。如图所示，ANTs 在所有数据集上始终表现出更优的性能。在 TUS 数据集上，ANTs 的表现最佳，其得分高于其他数据集。所有数据集中，表现第二好的系统是 ER。此外，与另外两个数据集不同，我们观察到在 $l=3$ 处，ANTs 和 ER 在 Ugen-v2 数据集上的 SNM 有所下降。进一步调查发现，可并集表数量稀少是导致这些现象的原因。



(a) Santos (b) Santos

(c) TUS (d) TUS

(e) Ugen-v2 (f) Ugen-v2

图 3：SSNM（左）和 SNM（右）在各数据集上的系统性能，$l \in [2, 10]$。Y 轴已截断。

简化句法新颖度度量。与 SNM 相比，SSNM 提供的信息较少，因为它只衡量系统检索新颖表的能力，而不衡量其排序能力。图 3（左列 (a)、(c)、(e)）展示了各数据集在所有查询上的平均 SSNM。ANTs 在 TUS 和 Santos 上再次优于其他系统。鉴于其性能稳定，ER 排名

表 II：Ugen v2 查询的平均新颖度分数。

	l	Starmie	GMC	GMM	ER	SemNov	ANTs
nscore(T)	2	0.0	0.004	0.3702	0.3704	0.3829	**0.3900**
	3	0.0073	0.0065	0.1816	0.2383	0.2412	**0.2474**

第二，位居系统中的第二名。随着 l 增加，所有方法的结果趋于一致；鉴于每个查询的可并集表数量有限（最多 20 个），这是预期的结果。对于 Ugen-v2 数据集，由于可并集表数量稀少，大多数系统的性能都明显下降。

搜索新颖性得分（nscore）。在我们的中型数据集（TUS 和 Santos）上计算搜索新颖性，即使使用 20 个中央处理器进行并行化，计算量仍然很大。因此，我们将搜索新颖性的计算限制在 $l \in [2, 3]$，并且仅专注于 Ugen-v2 数据集。

表 II 显示，ANTs 优于所有其他方法，取得了最高的平均新颖性。我们通过配对 t 检验评估了 $l = 2$ 和 $l = 3$ 的统计显著性。ANTs 相较于 Starmie、GMC 和 SemNov 的提升具有统计显著性（$p \leq 0.05$）。SemNov 排名第二，且在统计上显著优于 Starmie 和 GMC。ANTs 和 SemNov 的新颖度分数与 GMM 相当，未观察到统计显著差异。这表明，基于惩罚的方法取得的新颖度分数与可证明的$_1$ $_2$近似基线相当。

另一个重要观察是，就新颖度分数而言，ANTs 是整体表现最佳的系统（表 II）。其次是 SemNov 和 ER，它们在新颖度分数上竞争第二名，之后是 GMC，而 Starmie 的表现最差。这一排序与它们在 SNM、SSNM 和 Blatant-Duplicate 上的相对表现一致，表明我们的评估指标有效地衡量了预期目标。

附加指标。表 III 比较了所有基准测试中 ANTs 相对于基线方法的 $\mathcal{F}$ 值。受篇幅限制，我们仅报告 ANTs 在 $l = 10$达到相等或更高 $\mathcal{F}$ 值的查询平均占比。$l \in [2, 10]$ 的结果见附录中的表 VII。在所有 l级别上，ANTs 在 Santos 和 TUS 数据集上的表现始终不如 GMC；在 $l \in [2, 4]$方面，ANTs 在 Ugen-v2 数据集上的表现也始终不如 GMC，这凸显了 GMC 在实现其优化目标方面的有效性。然而，在所有数据集和 l级别上，ANTs 的表现始终优于 SemNov 和 ER，至少在 54.3% 的查询中取得了更高的性能。考虑到 GMC 的计算开销较大，当运行时间性能至关重要时，ANTs 比其他方法更高效、更有效。

表 III：ANTs 在 $l = 10$ 时达到不低于 $\mathcal{F}$ 的查询比例。≥ 50 值以斜体标示。

	Starmie	GMC	ER	SemNov
Santos	48.6%	2.9%	*54.3%*	*82.9%*
TUS	*66.7%*	2.4%	*73.8%*	*66.7%*
Ugen-v2	*73.7%*	*73.7%*	*84.2%*	*94.7%*

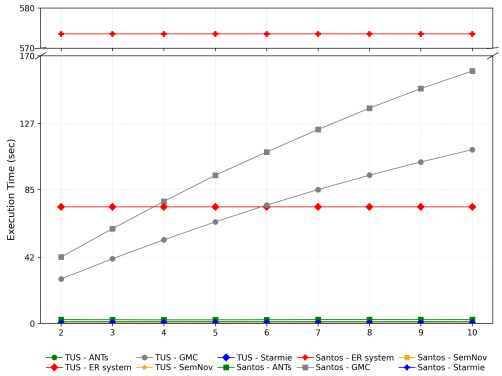


图 4 $l \in [2, 10]$：各数据集上的执行时间，按所有查询取平均值（）。Y 轴已截断。

可扩展性。我们比较了各种系统在重排序任务中的运行时间性能。在 Ugen-v2 上，所有系统的执行时间都可以忽略不计（≈ 0）。因此，我们在图 4 中报告了 TUS 和 Santos 上所有查询的平均运行时间。ANTs 和 SemNov 在这些数据集上的执行时间都很短，不到 2.4 秒。相比之下，GMC 和 ER 会产生显著的开销。这种显著的延迟使 GMC 和 ER 尤其不适用于数据湖上的交互式数据分析场景，因为低延迟响应对于实现有效的用户交互至关重要。

D. 与 DUST 的比较

在一篇即将发表的论文中，Khatiwada 等人 [34]研究了一个相关问题，并提出了 DUST 系统。给定一个查询表 Q，DUST 旨在找出 l 与 Q 以及彼此之间差异最大的可联合元组。与我们的方法类似，他们的方法建立在现有可联合搜索技术之上；但与我们不同的是，该方法不会对表进行重排序。相反，它在元组级进行操作，返回从多个可联合表中选出的一组最多样化的元组。此外，Khatiwada 等人 [34] 关注语义不相似度，而我们的工作旨在开发一种表重排序方案，以最大化句法不相似度（同时保持较高的可联合性或语义相似度）。

为了使 DUST 适用于我们的问题，我们通过计算该表的搜索新颖性得分，并将其与 ANTs 返回的排名第一的表的搜索新颖性得分进行比较，来评估其结果（一个多样化元组的集合，我们将其建模为单个表）的新颖性。为此，我们找出 ANTs 排名最高的表的大小（l），并运行 DUST 以找出最多样化的 l 个元组。我们预计 DUST 的表现会优于 ANTs，因为它从许多表中提取元组。我们的实验在 Ugen-v2 上运行，结果表明，尽管 DUST 获得的新颖度分数略高于 ANTs，但它只有在检索了多得多的表和元组并付出显著更高的运行时间后才能做到这一点（表 IV）。我们报告了所有查询中 ANTs 排名第一的表的平均大小，以及获取的元组总数（以及搜索中使用的表数量（#Acq.tup）。对于 ANTs，#Acq.tup 就是排名第一结果的大小，并且是 k 表中元组数量之和

表 IV：展示了新颖度分数、获取的元组和表的数量、最终新颖元组的数量以及执行时间。最佳值以**粗体**显示。

	nscore(T)	#获取元组数	#新颖元组数	#获取表数	执行时间（秒）
ANTs	0.4614	**23**	**23**	**1**	$\approx$ **0**
DUST	**0.4966**	416	**23**	5	101

DUST 正是从该表中提取其多样化元组。因此，在存在元组获取成本的数据市场场景中，或在对延迟敏感的场景中，ANTs 提供了更好的成本效益权衡；而当新颖性是唯一目标且资源不受限制时，DUST 更为可取。

E. 消融研究：组成技术与参数

我们报告了针对方法关键组成部分开展的三项消融研究的主要发现。

对齐质量：研究发现，在 SSNM 的所有（系统 X，数据集 Y）设置中，使用自动对齐算法 [34]（而非真实对齐）会造成负面影响的比例为 92%；在 SNM 的所有设置中，这一比例为 89%。然而，即使对齐质量较差，ANTs 在 SSNM 和 SNM 的所有可能（l，数据集 Y）设置中，仍分别有 48% 和 89% 的设置优于其他 NTS 方法。

搜索结果新颖性：接下来，我们研究初始搜索结果的新颖性如何影响新颖性感知重排序器的效果。我们的实验表明，就 SSNM 而言，初始结果的冗余增加会降低 ANTs 在所有数据集上的性能；SNM 也呈现出类似趋势。

超参数：对域大小阈值 s 和惩罚程度 b 的消融结果进行分析后发现，在所考察的数据集上，我们的方法具有鲁棒性，不受这些参数选择的影响：SNM 和 SSNM 在测试值范围内仅表现出有限的变化。因此，ANTs 对 s 和 b的具体取值并不敏感：即使这些参数未设为经验上的最优值，其性能仍保持稳定，$s = 10$ 和 $b \in 4$，5适用于所有数据集。

VIII. ANTs用于下游任务

我们使用可免费获取的 IMDb Extensive Dataset，评估 ANTs 对下游评分预测任务的影响。[11] 我们比较了构建训练数据的三种配置：(i) 基线，不应用数据发现，仅使用查询表；(ii) Starmie，使用 Starmie 返回的表扩充查询表；(iii) ANTs，使用 Starmie 返回并经 ANTs 重排序的表扩充查询表。我们构建回归模型，根据电影元数据预测用户平均评分（avg_vote）。[12] IMDb 电影表被随机划分为训练集（80%）和测试集（20%）。从训练集中随机选取 1% 的元组，并将它们划分为三个不同的查询表，每个表均保留所有预测属性（特征）。随后，我们使用剩余训练元组创建一个由可联合表组成的合成数据湖

[11]https://www.kaggle.com/datasets/simhyunsu/imdbextensivedataset[12]此任务设计是对用于展示其他数据发现方法有效性的任务的修改版本 [11]。

，方法遵循 Nargesian 等人 [1]。对于每个查询，我们还创建了可联合表的一个稀释变体（稀释度为 0.4），并手动将查询表插入数据湖。我们使用三种标准回归器：梯度提升机（LGBM）、随机森林（RF）和 XGBoost（XGB），并采用标准回归指标（R^2 和 RMSE）进行评估。[13] 使用可联合表的随机子集训练回归器后，我们发现，对于所有 $k \in [2, 10]$，无论采用 R^2 还是 RMSE，LGBM 的表现始终优于其他模型；因此，我们采用 LGBM 来比较表发现策略。在稀释数据集上，由于训练数据集中必然存在冗余，对于所有 k，无论采用 R^2 还是 RMSE，ANTs 的表现始终优于 Starmie 和基线。配对 t 检验（$p \leq 0.05$）表明，相较于基线的改进具有显著性；除 k=5 外，相较于 Starmie 的所有改进在统计上均显著。在非稀释数据集上，无论采用 R^2 还是 RMSE，ANTs 的表现都显著优于基线，但 ANTs 与 Starmie 之间未观察到显著差异。总的来说，我们得出结论：存在冗余时，使用 ANTs 对 Starmie 的结果进行重排序，始终有利于下游机器学习任务。在非稀释设置下，重排序步骤不会带来负面影响；因此，即使数据集的冗余程度未知，也可以放心地始终应用新颖性感知重排序器。

IX. 相关工作

表发现。给定查询表和扩展策略，从数据湖中发现有用的表近年来得到了广泛研究 [12], [35], ，其中可连接扩展和可并联合扩展最受关注。该问题的多个方面都已得到探讨，包括可扩展性和准确性。聚焦于可联合表搜索，Nargesian 等人 [1] 将共享可并联合属性的两张表定义为可联合表。为评估属性的可联合性，他们使用属性值、属性值到本体的映射，以及属性值的语义嵌入。D3L 扩展了属性可联合性的概念 [16]。Khatiwada 等人 [2] 通过对属性之间的关系进行建模，进一步改进了可联合表搜索。鉴于属性理解对于发现可联合表至关重要，近期研究日益关注如何增强属性表示。Hu 等人 [36] 和 Fan 等人 [11] 利用上下文信息，构建更丰富、更有效的属性表示。

新颖数据发现。大量文献探讨了如何在各种类型的系统中检索新颖结果，并根据其支持的用户查询形式进行分类，包括直接用户查询（即席信息检索 IR） [13], [37], 、间接用户查询（推荐系统）、以准 SQL 语句表示的查询 [38], 或以表表示的查询 [34]。一些系统还允许用户表达对结果质量与检索速度之间权衡的偏好 [38], [39]。如需全面了解信息检索和推荐系统中新颖性搜索的相关工作，读者可参阅 Wu等人 [37]。

元组相似度。我们的方法基于查询表与候选可联合表中元组之间的句法相似度。先前工作已提出多种在其他情境下比较元组的技术。在缺乏模式信息和其他元数据的情况下，已有框架用于计算数据库实例之间的相似度，即使它们包含缺失数据（即空值） [19]。Glavic等人 [19] 提出了一种基于数据库实例元组之间匹配的相似度度量。他们区分不同类型的空值，并在构建该度量时加以利用。相比之下，我们的方法假定所有空值都被统一处理。

X. 结论与未来工作

我们解决了从数据湖中识别新颖可合并表的问题。为此，我们提出了任何评分机制在根据表的新颖性进行排序时应满足的两个一般性质。我们提出并比较了若干新颖表搜索（NTS）方法。第一种方法 ANTs 根据相关性（语义相似度）和多样性（句法不相似度）对表进行排序，也就是说，检索到的表应尽可能与查询相关，同时结果集应尽可能多样。ANTs 是一种新的、可扩展的近似算法，我们将其与 GMC 进行比较；GMC 是一种已知较慢的查询多样化近似算法，使用与 ANTs 相同的相关性和多样性函数。SemNov 使用语义不相似度（具体而言是表嵌入之间的距离）而非句法不相似度。第四种方法 ER 使用基于实体的不相似度，即将实体重叠（在应用实体解析之后）的倒数用作表的多样性。我们的实验表明，ANTs 优于其他 NTS 方法。我们还展示了它对下游数据密集型任务的积极影响，并对其组成部分进行了若干消融研究。这项工作的若干方面仍有待未来探索。

查询表质量。如果查询表包含质量较差的数据，搜索算法可能缺乏足够的信息来识别真正新颖的表。受到信息检索查询重构技术的启发 [40],，提升查询表质量的一种方法是利用预言机（例如大语言模型）生成的假设元组或属性来扩充查询。我们计划研究查询扩展对 NTS 的影响。新颖性感知表嵌入。在这项工作中，我们通过属性嵌入的相似度计算相关性，并将其作为表可合并性的代理指标。未来的一个研究方向是将多样性（或新颖性）直接纳入用于生成属性嵌入的目标函数，从而通过统一的端到端流程解决 NTS 问题。

基准测试。我们实验中使用的基准测试最初并非为 NTS 问题而设计，因此它们

[13]我们使用 scikit-learn 的 API https: //scikit-learn.org 实现并评估了所有模型。

可能无法充分体现我们框架的能力。作为未来工作的一部分，我们计划开发专门针对 NTS 任务的新开放基准。

XI. 致谢

我们感谢加拿大卓越研究主席（CERC）Program 提供的支持。我们感谢加拿大卓越研究主席（CERC）给予支持。

XII. 人工智能生成内容声明

本文所呈现的所有构想、方法和结果均由作者独立构思并完成。人工智能工具并未参与任何科学构想、方法、分析或结论的形成。作者偶尔使用人工智能写作助手来获取措辞建议，并纠正拼写或语法错误。

参考文献

[1] F. Nargesian、E. Zhu、K. Q. Pu 和 R. J. Miller，“开放数据上的表联合搜索”，*Proc. VLDB Endow.*，第 11 卷，第 7 期，第 813–825 页，2018 年。[2] A. Khatiwada、G. Fan、R. Shraga、Z. Chen、W. Gatterbauer、R. J. Miller 和 M. Riedewald，“Santos：基于关系的语义表联合搜索”，*Proc. ACM Manag. Data*，第 1 卷，第 1 期，第 1–25 页，2023 年。[3] A. S. R. Santos、A. Bessa、C. Musco 和 J. Freire，“用于相关数据集搜索的基于草图的索引”，载于 *ICDE*，2022 年，第 2928–2941 页。[4] S. Castelo、R. Rampin、A. S. R. Santos、A. Bessa、F. Chirigati 和 J. Freire，“Auctus：用于数据发现与增强的数据集搜索引擎”，*Proc. VLDB Endow.*，第 14 卷，2794，2021 12，2791 no. pp. – 。[5] Y. Dong、K. Takeoka、C. Xiao 和 M. Oyamada，“数据湖中的高效可连接表发现：一种基于高维相似度的方法”，载于 *ICDE*，2021 年，第 456–467 页。[6] C. Koutras、G. Siachamis、A. Ionescu、K. Psarakis、J. Brons、M. Fragk- oulis、C. Lofi、A. Bonifati 和 A. Katsifodimos，“Valentine：评估用于数据集发现的匹配技术”，载于 *ICDE*，2021 年，第 468–479 页。[7] Y. Zhang 和 Z. G. Ives，“为交互式数据科学在数据湖中查找相关表”，载于 *SIGMOD*，2020 年，第 1951–1966 页。[8] D. Brickley、M. Burgess 和 N. F. Noy，“Google 数据集搜索：在开放网络生态系统中构建数据集搜索引擎”，载于 *WWW*，2019 年，第 1365–1375 页。[9] E. Zhu 、 D. Deng 、 F. Nargesian 和 R. J. Miller，“JOSIE：用于在数据湖中查找可连接表的重叠集合相似性搜索”，载于 *SIGMOD*，2019 年，第 847–864 页。[10] R. J. Miller、F. Nargesian、E. Zhu、C. Christodoulakis、K. Q. Pu 和 P. Andritsos，“让开放数据透明：开放数据上的数据发现”，*IEEE Data Eng. Bull.*，第 41 卷，第 2 期，第 59–70 页，2018 年。[11] G. Fan、J. Wang、Y. Li、D. Zhang 和 R. J. Miller，“利用上下文化的列级表示学习，在数据湖中进行语义感知数据集发现”，*Proc. VLDB Endow.*，第 16 卷，第 7 期，第 1726– 1739 页，2023 年。[12] A. Chapman、E. Simperl、L. Koesten、G. Konstantinidis、L. Ib´a˜nez、E. Kacprzak 和 P. Groth，“数据集搜索：综述”，*Proc. VLDB Endow.*，第 29 卷，第 1 期，第 251–272 页，2020 年。[13] C. L. A. Clarke、M. Kolla、G. V. Cormack、O. Vechtomova、A. Ashkan、S. B¨uttcher 和 I. MacKinnon，“信息检索评估中的新颖性与多样性”，载于 *SIGIR*，2008 年，第 659–666 页。[14] S. Mehta、M. Dawande、G. Janakiraman 和 V. S. Mookerjee，“如何出售数据集？数据变现的定价政策”，*Inf. Syst. Res.*，第 32 卷，第 4 期，第 1281–1297 页，2021 年。[15] J. Zhang、Y. Bi、M. Cheng、J. Liu、K. Ren、Q. Sun、Y. Wu、Y. Cao、R. C. Fernandez、H. Xu、R. Jia、Y. Kwon、J. Pei、J. T. Wang、H. Xia、L. Xiong、X. Yu 和 J. Zou，“数据市场综述”，*CoRR*，第 abs/2411.07267 卷，2024 年。[16] A. Bogatu、A. A. A. Fernandes、N. W. Paton 和 N. Konstantinou，“数据湖中的数据集发现”，载于 *ICDE*，2020 年，第 709–720 页。[17] M. R. Vieira、H. L. Razente、M. C. N. Barioni、M. Hadjieleftheriou、D. Srivastava、C. T. Jr. 和 V. J. Tsotras，“关于查询结果多样化”，载于 *ICDE*，2011 年，第 1163–1174 页。

[18] K. Pal、A. Khatiwada、R. Shraga 和 R. J. Miller，《ALT-GEN：使用大语言模型对表联合搜索进行基准测试》，载于 VLDB Workshops 论文集，2024。[19] B. Glavic 、 G. Mecca 、 R. J. Miller 、 P. Papotti 、 D. Santoro 和 E. Veltri，《不完整数据库实例的相似度度量》，载于 EDBT，2024 年，第 461–473 页。[20] E. F. Codd，《扩展数据库关系模型以捕获更多含义》，《ACM 数据库系统汇刊》，第 4 卷第 4 期，第 397–434 页，1979 年。[21] M. Yu、G. Li、D. Deng 和 J. Feng，《字符串相似性搜索与连接：综述》，《计算机科学前沿》，第 10 卷第 3 期，第 399–417 页，2016 年。[22] J. Lin，《基于香农熵的散度度量》，《IEEE 信息论汇刊》，第 37 卷第 1 期，第 145–151 页，1991 年。[23] I. S. Dhillon、S. Mallela 和 R. Kumar，《用于层次文本分类的增强词聚类》，载于 SIGKDD，2002 年，第 191–200 页。[24] J. Kieffer，《信息论要素（Thomas M. Cover 和 Joy A. Thomas）》，《SIAM 综述》，第 36 卷第 3 期，第 509–511 页，1994 年。[25] S. S. Ravi、D. J. Rosenkrantz 和 G. K. Tayi，《分散问题的启发式算法与特殊情形算法》，《运筹学》，第 42 卷第 2 期，第 299–310 页，1994 年。[26] Z. Moumoulidou、A. McGregor 和 A. Meliou，《公平性约束下的多样化数据选择》，载于 ICDT，第 186 卷，2021 年，第 13:1–13:25 页。[27] H. Iida、D. Thai、V. Manjunatha 和 M. Iyyer，《TABBIE：表格数据的预训练表示》，载于 NAACL-HLT，2021 年，第 3446–3456 页。[28] P. Konda、S. Das、P. S. G. C.、A. Doan、A. Ardalan、J. R. Ballard、H. Li、F. Panahi、H. Zhang、J. F. Naughton、S. Prasad、G. Krishnan、R. Deep 和 V. Raghavendra，《Magellan：迈向构建实体匹配管理系统》，《VLDB 会议录》，第 9 卷第 12 期，第 1197–1208 页，2016 年。[29] R. Guo、P. Sun、E. Lindgren、Q. Geng、D. Simcha、F. Chern 和 S. Kumar，《利用各向异性向量量化加速大规模推理》，载于 ICML，第 119 卷，2020 年，第 3887–3896 页。[30] Y. N. Silva、W. G. Aref、P. Larson、S. Pearson 和 M. H. Ali，《相似性查询：概念评估、转换与处理》，《VLDB 会议录》，第 22 卷第 3 期，第 395–420 页，2013 年。[31] S. Thirumuruganathan、H. Li、N. Tang、M. Ouzzani、Y. Govind、D. Paulsen、G. Fung 和 A. Doan，《用于实体匹配中阻塞的深度学习：设计空间探索》，《VLDB 会议录》，第 14 卷第 11 期，第 2459–2472 页，2021 年。[32] T. Sakai，《bpref 的替代方案》，载于 SIGIR，2007 年，第 71–78 页。[33] A. Khatiwada、R. Shraga、W. Gatterbauer 和 R. J. Miller，《数据湖表集成》，《VLDB 会议录》，第 16 卷第 4 期，第 932–945 页，2022 年。[34] A. Khatiwada、R. Shraga 和 R. J. Miller，《多样化可联合元组搜索：数据湖中的新颖性驱动发现》，载于 EDBT，2026 年，第 42–55 页。[35] G. Fan、J. Wang、Y. Li 和 R. J. Miller，《数据湖中的表发现：现状与未来方向》，载于 SIGMOD，2023 年，第 69–75 页。[36] X. Hu、S. Wang、X. Qin、C. Lei、Z. Shen、C. Faloutsos、A. Katsifodimos、G. Karypis、L. Wen 和 P. S. Yu，《利用表格表示学习自动进行表联合搜索》，载于 ACL，2023 年，第 3786–3800 页。[37] H. Wu、Y. Zhang、C. Ma、F. Lyu、B. He、B. Mitra 和 X. Liu，《搜索与推荐中的结果多样化：综述》，《IEEE 知识与数据工程汇刊》，第 36 卷第 10 期，第 5354–5373 页，2024 年。[38] H. Mirzaei 和 D. Rafiei，《带偏好的表联合搜索》，载于 VLDB Workshops 论文集，2023 年。[39] M. R. Vieira、H. L. Razente、M. C. N. Barioni、M. Hadjieleftheriou、D. Srivastava、C. T. Jr. 和 V. J. Tsotras，《DivDB：一种用于查询结果多样化的系统》，《VLDB 会议录》，第 4 卷第 12 期，第 1395–1398 页，2011 年。[40] B. Kassaie、A. Kane 和 F. W. Tompa，《在数学感知搜索引擎中利用查询重构与倒数排名融合》，载于 DocEng，2025 年，第 7:1–7:10 页。[41] M. R. Garey 和 D. S. Johnson，《计算机与难解性：NP完全性理论指南》，1979 年。[42] Y. Li 、 J. Li 、 Y. Suhara 、 A. Doan 和 W. Tan，《利用 Transformer 实现有效的实体匹配》，《VLDB 会议录》，第 32 卷，1235，2023 年 6，1215 页 – 。[43] Y. Li、J. Li、Y. Suhara、A. Doan 和 W. C. Tan，《利用预训练语言模型进行深度实体匹配》，CoRR，第 abs/2004.00584 卷，2020 年。[44] U. Brunner 和 K. Stockinger，《基于 Transformer 架构的实体匹配——数据集成领域的进步》，载于 EDBT，2020 年，第 463–473 页。[45] F. Neuhof、M. Fisichella、G. Papadakis、K. Nikoletos、N. Augsten、W. Nejdl 和 M. Koubarakis，《过滤技术的开放基准测试》

在实体解析领域》，《*VLDB* 会议录》，第 33 卷，1696，2024 年 5，1671 页 – 。[46] M. F. Porter，《一种用于后缀剥离的算法》，《*Program*》，第 14 卷第 3 期，第 130–137 页，1980 年。[47] P. J. Rousseeuw，《轮廓系数：用于解释和验证聚类分析的图形辅助工具》，《计算与应用数学杂志》，第 20 卷，第 53–65 页，1987 年。[48] K. Wagstaff、C. Cardie、S. Rogers 和 S. Schr¨odl，《利用背景知识进行约束 k-means 聚类》，载于*ICML*，2001 年，第 577–584 页。[49] B. Babaki，《Cop-kmeans 1.5 版》，2017 年。

附录

A. 证明

本节给出两个证明，以支持正文中的定理。

定理 1. 搜索新颖度分数（定义 *8*）满足明显重复公理和稀释公理。

证明 *1*： 令 T 为按照定义 8 所述方法构造的搜索结果表。对于每个元组 $t \in T$，我们将根据定义6计算出的元组新颖度纳入结果的整体新颖度分数中，如公式（2）所定义。根据构造，所有新添加的元组都来自查询表，而查询表中已经存在完全相同的对应元组，因此：

i）对于新引入的元组，所有由此产生的 N（t）均为零，因为所有元组对中都包含一个查询表中已有的完全相同的对应元组 T：公式（2）中的分母（y）增大，而分子保持不变。ii）对于已有元组，实际上我们添加的是已计入的元组，因此这不会影响定义6中的最小值，却会增加定义7中的元组数量（即，更大的 y）。

因此，添加查询元组会导致 nscore 降低。同样，我们可以证明 $nscore$ 函数（定义 8）满足稀释公理。

定理 2. 在定义 *9* 中寻找最优的 R 是 *NP* 困难问题。

证明 *2*： 如果将集合 S 表示为一个完全图，那么该问题类似于具有属性 Π和大小 l [41]的诱导子图问题，其中 Π 在我们的情况下要求大小为 l 的子图与所有其他相同大小的子图相比具有最大 nscore。由于验证属性Π 需要指数时间，因此我们的问题是 NP 困难问题。

B. 詹森-香农散度

我们简要回顾詹森-香农散度 [23]。设属性 B_j 和 A_i 为离散随机变量，其取值来自集合 $\mathcal{D}$，相应的概率分布为 $\mathbb{B}_i$ 和$\mathbb{A}_j$。两个分布 $\mathbb{B}_i$ 和 $\mathbb{A}_j$ 之间的相对熵或 Kullback-Leibler（KL）散度计算如下：

$$\mathrm{KL}(\mathbb{B}_i, \mathbb{A}_j) = \sum_{v \in \mathcal{D}} \mathbb{B}_i(v) \times \log \frac{\mathbb{B}_i(v)}{\mathbb{A}_j(v)}$$

库尔贝克-莱布勒（KL）散度存在若干问题，因此不适用于我们的使用场景。它不是一个恰当的距离度量，因为它不对称且不满足三角不等式。此外，当存在缺失值时，它没有上界：如果 $\mathbb{B}_i(v) = 0$ 且 $\mathbb{A}_j(v) \neq 0$, $KL(\mathbb{B}_i, \mathbb{A}_j) = -\infty$，这使它不适用于我们的问题，因为任一概率分布中都可能缺少概率质量。为了解决这些局限，我们考虑使用詹森-香农散度（JSd）。它源自 KL 散度，但具有一些理想性质：它是对称的，取值范围在 $[0, 1]$, 以内，并且可以作为恰当的距离度量：

$$\mathrm{JSd}(\mathbb{B}_i, \mathbb{A}_j) = \frac{1}{2} \times (\mathrm{KL}(\mathbb{B}_i, \mathbb{M}), \mathrm{KL}(\mathbb{A}_j, \mathbb{M}))$$

其中 $\mathbb{M} = \frac{\mathbb{B}_i + \mathbb{A}_j}{2}$ 是 $\mathbb{B}_i$ 和 $\mathbb{A}_j$的混合分布。两个概率分布之间的 Jensen-Shannon 距离（JSD）是 JSd 的平方根：

$$\text{JSD}(\mathbb{B}_i, \mathbb{A}_j) = \sqrt{\text{JSd}(\mathbb{B}_i, \mathbb{A}_j)}$$

C. 消融实验：数据集稀缺性的影响

在本节中，我们研究数据集稀缺性对结果的影响。

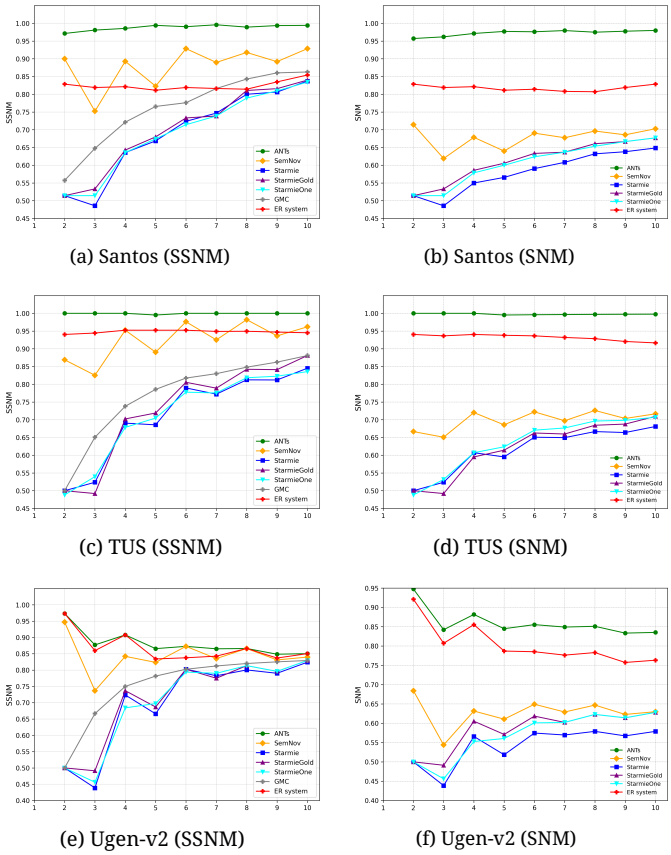
1）句法新颖性度量：进一步检查后发现，在 Ugen-v2 小型版数据集中，对于 ANTs 在 $l = 3$ 表现不佳的查询，其中 80% 在真实标注中仅有四个可联合表，其中两个是查询表的明显重复项（包括其稀释版本）。这些低质量匹配会对 SNM 分数产生负面影响。可联合表的稀缺是观察到的性能下降的原因之一。为了检验这一假设，我们进一步将 Ugen-v2 小型版数据集限制为至少有 12 个可联合表的查询，结果呈现出与其他数据集一致的趋势（图 9b）。

2）简化句法新颖度度量：同样，在 Ugen-v2 小型版数据集中，观察到 ANTs 的性能出现异常下降，这归因于可联合表数量有限。为验证这一假设，我们报告了至少有 12 个可联合表的查询的 SSNM 分数。图 9a 所示结果支持了我们的假设，并证实了数据稀疏性对性能的影响。

D. 面向新颖表搜索的实体解析

ANTs 采用基于属性的策略来降低新型表搜索问题（定义 9）的时间复杂度。另一种方法是考虑基于元组的近似方法。例如，可以应用实体解析（也称为记录链接或实体匹配）技术来估计数据湖中的查询表与候选表之间的重叠度，即相同实体的数量，然后对表进行排序，使重叠较少的表获得更高的分数。接下来，我们研究实体解析技术在解决新型表搜索问题中的应用。

实体解析侧重于确定两个元组是否指向同一个现实世界实体。在这一研究方向中，研究人员开发了相关方法，用于从两个或更多数据源（例如表 [28], [31], [42]–[44]）中识别匹配实体对。穷举式实体解析相对于元组数量具有二次时间复杂度。为提高可扩展性，通常采用阻塞技术，将相似元组聚类，并将比较限制在每个阻塞内的候选对，从而有效降低匹配阶段的计算开销。在执行实体解析时，有必要确定应采用适当的阻塞和匹配方法。当查询表到达时，首先会经历阻塞阶段，在该阶段识别查询表与每个可并集表之间的相似元组，并将其分组到阻塞中。随后，匹配步骤仅限于落在同一阻塞内的元组对。



(a) Santos (SSNM) (b) Santos (SNM)

(c) TUS (SSNM) (d) TUS (SNM)

(e) Ugen-v2 (SSNM) (f) Ugen-v2 (SNM)

图 5：在 Santos、TUS 和 Ugen-v2 小型版数据集上对比 SNM（右列）与 SSNM（左列）。

阻塞。最近， [45] 针对多个广为人知的数据集，对各种阻塞技术开展了广泛的实验研究，并从多个维度比较了它们的性能。我们根据他们的研究结果选择阻塞方法，以确定最适合我们设置的技术。在选择阻塞技术时，我们考虑以下关键特征。我们倾向于采用无需领域知识或标注数据的自监督阻塞策略，因为在数据湖环境中，由于缺少元数据且元组规模庞大，这两者都不切实际。因此，我们采用与模式无关的设置，不依赖于确定适合阻塞的属性，而是将每个元组表示为词元。最重要的是，该技术必须具有较高的计算效率，尤其是在对可并集表搜索结果进行重排序时，因为这本质上是一个在线过程。例如，我们采用了 DeepBlocker 框架中的自编码器组件，先前研究已证明它效果良好 [31],[45]。这种方法很适合我们的设置，因为它不需要标注数据。然而，尽管它能实现高精确率，我们发现其计算成本很高。在 Santos 数据集上的测试中，该方法速度极慢，因此被排除在我们的实验之外。具体而言，我们遇到了两个主要问题：由于内存消耗过高，该过程始终需要超过我们可用的 512

GB 内存，而且运行速度很慢，单个查询的阻塞步骤需要两个多小时才能完成，因此我们不得不中止执行。Neuhof 等人提出的最有效且高效的方法 [45] 是随机连接，但我们找不到其公开可用的实现。此外，我们重点关注无需复杂微调和配置的技术。综合考虑这些因素，我们将分析范围限定为两种具有公开可用实现的阻塞技术：

1) K 近邻连接（kNN – 连接） [30] kNN –连接速度极慢；运行 16.85 小时后，我们终止了 KNN（k = 5）任务。此时，该任务在 Santos 数据集上仅处理了四张查询表（平均每张表耗时 4.2 小时）。
2) ScaNN [29] 该技术速度最快，所需的参数调优最少，而且在现有实现中内存占用较低，因此我们选择了它。

E. 归一化

我们介绍了在研究的两个阶段对属性应用的句法归一化流程。在域大小估计和句法相似度计算之前，我们会对属性值应用此归一化流程。请注意，在 Starmie 模型的微调或推理阶段，并未执行属性归一化。

由于没有关于表如何创建的信息，我们对所有表和属性一视同仁；因此，归一化函数的语义在所有属性中保持一致。给定任意表 $T \in \mathcal{T}$（$A_1, \ldots, A_n$），对于每个元组 $t \in T$其中 $t = \langle v_1, \ldots, v_n \rangle$，以及对于每个 $v_k \in t$，均按以下顺序执行下列步骤：

i) 分词。 v_k 被分词，即使用空格、句点、下划线和连字符作为分隔符进行切分。我们使用了 NLTK 现成的正则表达式分词器。[14]ii) 词汇标准化。 随后将这些词元转换为小写，并使用词干提取进行处理 [46]。 iii)重构。最后，将归一化后的词元按原始顺序重新合并，并以空格字符作为分隔符。

例如，应用上述三个步骤后，字符串“IT-Hardware Purchases”被归一化为“it hardwar purchas”。归一化函数的输入是字符串，输出同样是字符串。在计算 Jensen-Shannon 距离时，我们将该属性表示为归一化字符串的多重集，以保留频率；然而，在计算杰卡德相似度时，我们将该属性表示为归一化字符串的集合。

F. 扩展结果

我们扩展了实验，纳入了另外两种基线方法：StarmieOne 和 StarmieGold。本节将

[14]https://www.nltk.org/api/nltk.tokenize.regexp.html

表 V：每个单元格显示各数据集中在前 l 个结果内产生明显重复项的查询所占的平均百分比（%），该平均值是在范围 $l \in [2, 10]$ 上并对各数据集中的全部查询计算得出的。

	StarmieStarmieGoldStarmieOneGMC ER SemNovAN					
Santos 99.4	99	99	81.9 8.6	**0**	**0**
TUS 100	100	100	99.5 **0**	**0**	**0**
Ugen-v2 100	100	100	98.8**43.6**	52.6	43.9

表 VI：每个单元格表示 Ugen v2 小型版各查询的平均搜索新颖性得分。 –

	Starmie StarmieGold StarmieOne GMC			ER	SemNov	ANTs
$l = 2$	0.0	0.0	0.0	0.0040 0.3715 0.3840 <style id='标识符		
$l = 3$	0.0074	0.0076	0.0074	0.0066 0.2387 0.2417 **0**		

重新审视先前报告的结果，并将这两种基线方法纳入比较。StarmieGold 与 Starmie 类似，但将其内置的对齐方式替换为基于真实标注的对齐。属性使用 Starmie 嵌入表示，并依据真实标注映射进行对齐。随后，将对齐属性对之间的余弦相似度之和作为可并集性得分，并据此对检索到的表进行排名。我们还引入了一种名为 StarmieOne 的变体，其可并集性得分采用余弦相似度的平均值，而非像 Starmie 那样取总和。分析结果表明，Starmie 的不同变体对评估指标几乎没有影响（见图 5 和图 6，以及表 V 和表 VI）。

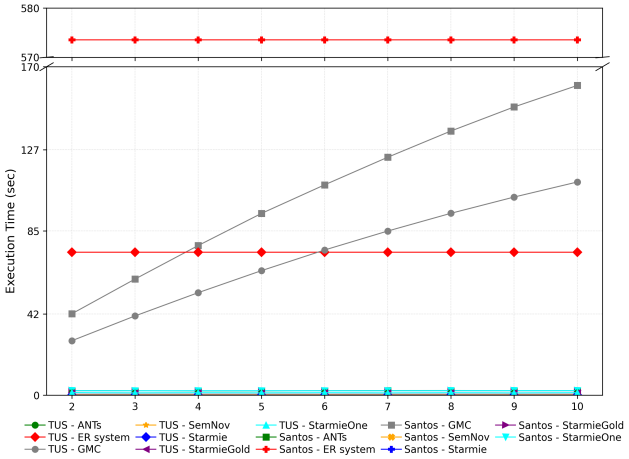


图 6：该图是图 4 的完整版本。

G. 完整结果

我们在表 VII 中报告了所有数据集和系统中所有 $l \in [2, 10]$ 的 $\mathcal{F}$ 值结果。

H. ANTs对比DUST

我们将 ANTs 与 Khatiwada 等人提出的一种近期可并集元组搜索技术 DUST 进行比较 [34]。我们通过计算搜索新颖性得分（定义 8）来评估 DUST 返回的结果有多新颖，并将其与 ANTs 的得分进行比较，

表 VII：对于 l 值从 2 到 10 的情况，ANTs 达到相等或更高 $\mathcal{F}$ 值的查询平均百分比。百分比超过 50% 的情况已突出显示。为便于阅读，我们省略百分号，表中将每个值写作 X，而非 X%。2 3 4 5 6 7 8 9 10

Santos									
Starmie	68.6			8.6 5			6	45.7	48.6
StarmieGold	68.6	48.6	5	2.9	42.9	42.9	40	40	40
StarmieOne	68.6	48.6	5	8.6	57		6	48.6	51.4
GMC	5.7	2.9	0	0	0	2.9	5.7	0	2.9
ER	65.7							60	54.3
SemNov	94.3								
TUS									
Starmie	69	69	66.7						
StarmieGold	69	69	69	61.9			2.9	35.7	38.1
StarmieOne	69	69	66.7				69	71.4	69
GMC	14.3	2.4	2.4	0		2.4	4.8	4.8	2.4	2.4
ER	71.4					81	78.6		
SemNov	83.3								
Ugen-v2 小型版									
Starmie	5.3	0	36.8	36.8	5				
StarmieGold	52.3	0	36.8	31.6	5				
StarmieOne	5.3	0	26.3	36.8	5				
GMC	0	0	31.6	36.8	5				
ER	78.9								
SemNov	78.9								

这些得分。如第七节所述，计算新颖度分数耗时极长，因此我们将此分析集中在 Ugen-v2 数据集上。对于这两个系统，我们将初始搜索结果中的可并集表数量设为 $k = 20$；对于每个查询，我们首先从 ANTs 获取最可并集的表，并将其元组数量记为 m。随后，我们要求 DUST 针对同一查询检索 m个最可并集的元组。我们使用 *nscore* 指标计算并比较结果表的新颖性。对所有查询取平均后，ANTs 返回的新颖性最高的表包含 $m = 23$ 个元组。表 IV 报告了每种方法的平均新颖度分数、达到该分数所需检索的表和元组总数的平均值、由此得到的新颖元组数量，以及相应的运行时间。尽管 DUST 的新颖度分数高于 ANTs（增益为 7.6%），但它平均需要检索比 ANTs 多五倍的表和十八倍的元组，且执行时间显著更长。因此，我们得出结论：在每个元组都附带获取成本的数据市场场景中，或在对延迟敏感的场景下，ANTs 是更好的选择。

I. ANTs机器学习任务

数据发现的目标是为下游任务准备数据。在本节中，我们将展示使用新颖性感知数据发现方法 ANTs 对机器学习模型性能的影响。我们任意选择了一项具有代表性的下游任务，即评分预测，以及一个可免费获取的数据集 IMDb Extensive Dataset。[15] 随后，我们在三种配置下训练模型：不进行数据发现，仅使用查询表（基线）；使用 Starmie 返回的表扩充查询表；以及使用 Starmie 返回并由 ANTs 重新排序的表扩充查询表。[16] 接下来，我们将详细介绍实验设置。

1) 数据准备：我们使用电影元数据（标题、年份、类型、导演、演员、时长、预算等），其中每部电影的平均用户评分由 avg_vote 属性给出。机器学习任务是训练一个回归模型，根据元数据特征预测 avg_vote。IMDb 电影表被随机划分为训练集（80%）和测试集（20%）。从训练集中随机选取 1% 的元组，并将其划分为三个不同的查询表，每个查询表都保留所有预测属性（特征）。随后，我们按照 Nargesian 等人 [1]提出的流程，使用剩余的训练元组构建合成数据湖。此过程在数据湖中生成三个查询表和 50 个可联合表。为确保训练数据集包含足够的冗余信息，我们使用原始数据湖以及稀释度为 0.4 的稀释变体开展实验；在该变体中，查询表也包含在数据湖中，因此总共有 101 个可联合表。我们采用回归模型的标准评估指标，即 R^2 分数（越高越好）和均方根误差分数（越低越好）。[17]

2）模型选择：首先，我们选择了三种知名的回归模型：Light Gradient Boosting Machine（LGBM）、随机森林（RF）和 XGBoost（XGB）。对于每个查询表，我们在从未稀释数据湖中随机选取的表的并集上训练这些模型（包括查询表本身），并报告三个查询的平均性能。如表 VIII 所示，在 k 的所有取值下，无论是 R^2 还是均方根误差，LGBM 的表现都始终优于其他模型。我们推测，这一性能增益源于 LGBM 内部处理 $\perp$的方式；在我们的设置中，这一方式尤其有利，因为结果表的构建（公式（3））会引入 $\perp$值。因此，我们选择 LGBM 作为回归模型，用于比较不同表发现技术的性能。

3）结果：表 IX 和 X 总结了 LGBM 在稀释和未稀释训练集上的性能。在稀释数据集上（其中冗余是必然存在的），对于k 的所有取值以及 R^2 和均方根误差这两项指标，ANTs 的表现始终优于 Starmie 和基线。我们使用配对 t 检验评估显著性（p 值 ≤ 0.05）；相较于基线的改进均具有显著性；相较于 Starmie 的改进在统计上均显著，k=5除外。在未稀释数据集上，对于 R^2 和均方根误差，ANTs 均显著优于基线，但 ANTs 与 Starmie 之间未观察到显著差异。

总体而言，我们得出结论：在存在冗余的情况下，使用 ANTs 对 Starmie 的结果进行重排序显然更为可取。即使在未稀释的设置中，重排序步骤也不会产生不利影响；因此，即使数据集的冗余程度未知，也可以始终安全地应用新颖性感知重排序器。

[15]https://www.kaggle.com/datasets/simhyunsu/imdbextensivedataset

[16]我们首先使用 Starmie 检索排名前 30 的表，然后应用 ANTs 对结果进行重排序。

[17]我们使用 scikit-learn https://scikit-learn。org 提供的标准 API 创建并评估了所有模型。

表VIII：随机数据选择从未稀释数据湖中选取的数据训练的三个模型的 R^2 和均方根误差。对于每个 k ，各模型在每项指标上的最佳值以**粗体**显示。

	R^2			RMSE	
k	LGBM	XGB	随机森林	LGBM	XGB	RF
2	**0.4447**	0.2714	0.2360	**0.9194**	1.0506	1.0752
3	**0.4671**	0.2790	0.2524	**0.9007**	1.0454	1.0638
4	**0.4577**	0.2694	0.2164	**0.9086**	1.0526	1.0878
5	**0.4554**	0.2702	0.2077	**0.9105**	1.0516	1.0933
6	**0.4565**	0.2697	0.2075	**0.9095**	1.0517	1.0930
7	**0.4366**	0.2155	0.1567	**0.9262**	1.0923	1.1307
8	**0.4322**	0.2553	0.2338	**0.9299**	1.0649	1.0802
9	**0.4335**	0.2530	0.2358	**0.9288**	1.0666	1.0788
10	**0.4352**	0.2559	0.2377	**0.9274**	1.0645	1.0774

J. 消融研究：对齐质量的影响

在本研究中，我们通过用自动生成的对齐替代真实对齐，考察对齐正确性对我们的方法效果的影响。借助近期一项相关工作 [34], 的代码库，我们采用基于聚类的方法，对齐查询表与数据湖表中的属性。[18] 与该工作 [34], 类似，我们的目标是形成若干个簇，每个簇恰好包含一个查询列，以及数据湖表中与之对应的对齐属性。根据领域知识，我们施加约束，禁止将同一张表中的两列聚到同一簇中（即与同一个查询属性对齐）。对于属性表示，我们采用列级 RoBERTa 模型（ [34]中表现最佳的模型），并使用轮廓系数 [47]确定最佳簇数。在对其代码稍作修改后，我们采用加入不可连接约束 [48]的 K-means 聚类，并使用余弦距离，取得了更好的结果。K-means 聚类算法的代码库可免费下载 [49]。我们将此对齐方法用于 GMC、SemNov、ER 和 ANTs。不过，Starmie 自带对齐流程。我们拥有所有数据集的对齐真实标注，因此能够报告对齐算法的精确率、召回率和 F 值，如表 XI 所示。正如预期，用自动对齐算法替代人工对齐会一贯损害性能。就 SSNM 而言（图 7），在 12 种（系统、数据集）配置中，有 11 种（92%）配置下 ANTs、SemNov 和 GMC 的性能有所下降。同样，在 89% 的配置下，ANTs 和 SemNov 的 SNM 性能也有所下降（图 8）。然而，即使在这些条件下，对于表数超过 12 的 Ugen-v2 查询，ANTs 在 SSNM 和 SNM 的（数据集、 l）设置中仍分别有 48% 和 89% 的情况下表现最佳。有趣的是，在 Santos 数据集上，ER 在 SSNM 和 SNM 上的性能均有所提升。

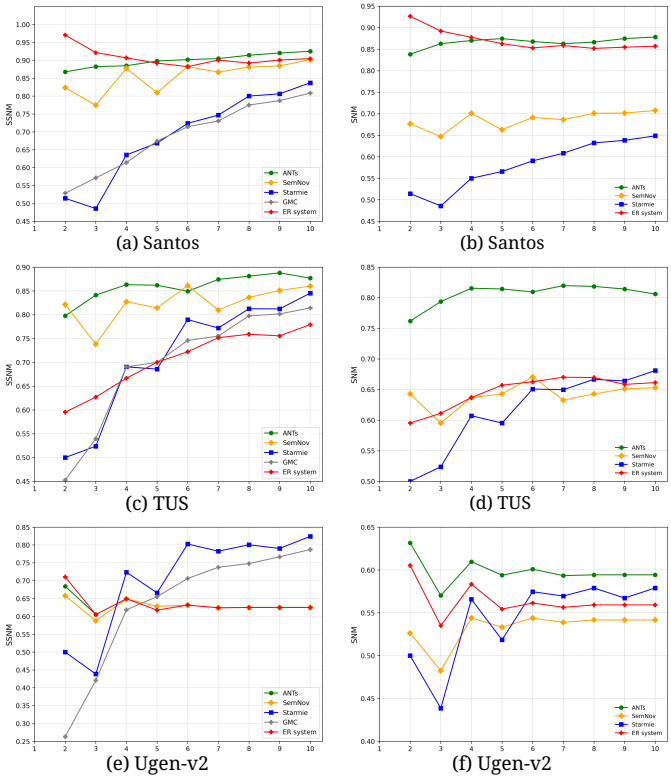


图 7：对于 $l \in [2, 10]$，SSNM（左）和 SNM（右）在各数据集上的系统性能（使用对齐算法）。纵轴已截断。

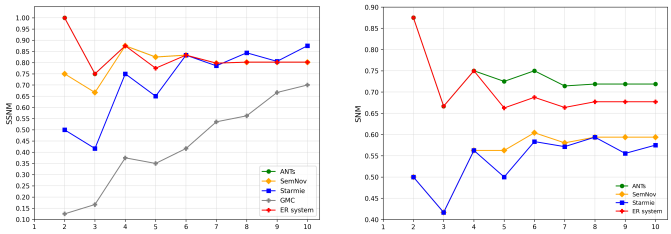


图 8：Ugen-v2 中包含 12 个以上可并集表（Ugen12）的查询结果；使用自动对齐。

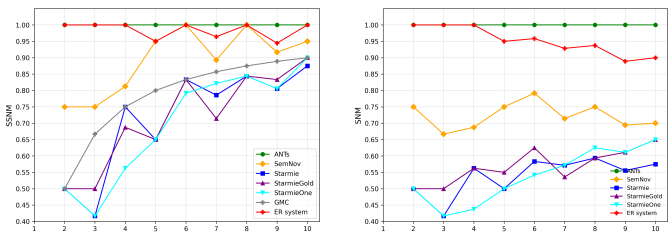


图 9：Ugen-v2 中包含 12 个以上可并集表（Ugen12）的查询结果。

[18]https://github.com/northeastern-datalab/dust

表 IX：在稀释数据集上按 k 计算的均方根误差（越低越好）和 R^2 （越高越好）。数值为三个查询表结果的平均值。增益表示 ANTs 相对于 Starmie 的相对变化。对于每个指标，每行的最佳值均以**粗体**标示。

	RMSE				R^2			
k	基线	Starmie	ANTs	Gain	基线	Starmie	ANTs	Gain
2	0.8975	0.8544	**0.8373**	−2.00%	0.4711	0.5206	**0.5396**	+3.65%
3	–	0.8553	**0.8324**	−2.68%	–	0.5196	**0.5450**	+4.89%
4	–	0.8471	**0.8364**	−1.26%	–	0.5288	**0.5406**	+2.24%
5	–	0.8507	**0.8380**	−1.49%	–	0.5248	**0.5389**	+2.69%
6	–	0.8512	**0.8383**	−1.52%	–	0.5242	**0.5385**	+2.72%
7	–	0.8524	**0.8378**	−1.71%	–	0.5229	**0.5391**	+3.10%
8	–	0.8504	**0.8381**	−1.45%	–	0.5252	**0.5388**	+2.59%
9	–	0.8514	**0.8384**	−1.53%	–	0.5240	**0.5384**	+2.74%
10	–	0.8532	**0.8405**	−1.49%	–	0.5219	**0.5361**	+2.71%

表 X：非稀释数据集上按 k 划分的均方根误差（越低越好）和 R^2 （越高越好）。数值为三个查询表结果的平均值。增益表示 ANTs 相对于 Starmie 的相对变化。每行中各项指标的最佳值以**粗体**显示。

	RMSE				R^2			
k	基线	Starmie	ANTs	Gain	基线	Starmie	ANTs	Gain
2	0.8975	0.8378	**0.8373**	−0.06%	0.4711	0.5391	**0.5396**	+0.09%
3	–	0.8359	**0.8324**	−0.42%	–	0.5412	**0.5450**	+0.71%
4	–	**0.8359**	0.8364	+0.06%	–	**0.5411**	0.5406	−0.10%
5	–	0.8380	**0.8380**	0.00%	–	**0.5389**	0.5389	+0.01%
6	–	**0.8375**	0.8383	+0.10%	–	**0.5394**	0.5385	−0.17%
7	–	**0.8378**	0.8378	0.00%	–	**0.5391**	0.5391	−0.003%
8	–	0.8388	**0.8381**	−0.08%	–	0.5380	**0.5388**	+0.14%
9	–	0.8392	**0.8394**	+0.02%	–	**0.5375**	0.5373	−0.04%
10	–	0.8396	**0.8418**	+0.26%	–	**0.5371**	0.5347	−0.44%

表 XI：对齐算法的有效性。

	Santos	TUS	Ugen-v2
精确率	0.80	0.88	0.96
召回率	0.81	0.62	0.95
F 值	0.78	0.70	0.95

表 XII：使用自动对齐时，每个数据集的查询中，在前 l 个结果（$l \in [2,10]$）内出现明显重复项的平均百分比。数值越低，性能越好。在所有表中，**粗体** 数字表示每行的最佳性能；带下划线的数字表示次佳性能。

	Starmie	GMC	ER	SemNov	ANTs
Santos	99.4%	89.2%	3.5%	4.1%	3.5%
TUS	100%	100%	**2.4%**	**3.2%**	**5.3%**
Ugen-v2	100%	69%	**84%**	85%	84.8%
Ugen12	100%	22.2%	**61%**	**61%**	**61%**
Ugen $l=2$	100%	100%	**47.4%**	**52.6%**	**47.4%**

K. 消融实验：超参数影响

我们的方法有两个关键参数：域大小阈值 s 和惩罚程度 b。在主要实验中，我们根据小规模参数探索的结果，以经验方式设定这些值，并在所有数据集上使用 $s = 20$ 和 $b = 1$。在这项消融实验中，我们研究它们对 ANTs 有效性的影响。我们采用两阶段（顺序）超参数调优策略：首先调整 s，然后在选定其取值的基础上，调整 b。第一阶段，我们将惩罚程度固定为 $b = 1$，并报告三项基准测试中 $s \in \{5, 10, 15, 20, 25, 30, 35\}$的 SNM 和 SSNM，这些指标是对所有查询和所有 $l \in [2, 10]$ 取平均所得。如图 10a 和 10b 所示，使 SNM 和 SSNM 达到最佳的 s 取值，Santos 为 10，TUS 为 5、10 或 15，Ugen-v2 为 10、15 或 20。这证实了 s应根据数据来选择。接下来，我们将$s = 10$固定为其在所有数据集上的最优值，并报告惩罚程度为 $b \in [1, 10]$时的结果。如图 10c 和 10d 所示，对于 TUS，SNM 和 SSNM 均在 $b \in [1, 5]$时达到最大值。在 Santos 上，SNM 相对稳定，对 b 的敏感度低于 SSNM；两项指标均在 $b \in [4, 10]$时达到最大值。对于 Ugen-v2，任意 $b \in [2, 10]$, 时 SNM 均达到最大值，而 SSNM 不会受到 b的显著影响。

L. 消融实验：初始搜索系统的影响

在本节中，我们研究初始搜索结果的新颖性如何影响我们的新颖性感知重排序器的有效性。具体而言，我们旨在回答以下问题：作为初始结果中冗余程度的代理指标，输入ANTs的稀释表与非稀释表的比例会如何影响最终的 *SSNM* 和 *SNM* 分数？为此，我们将初始搜索结果替换为合成的随机生成结果集，并控制假设的可合并表搜索系统返回的稀释表比例。

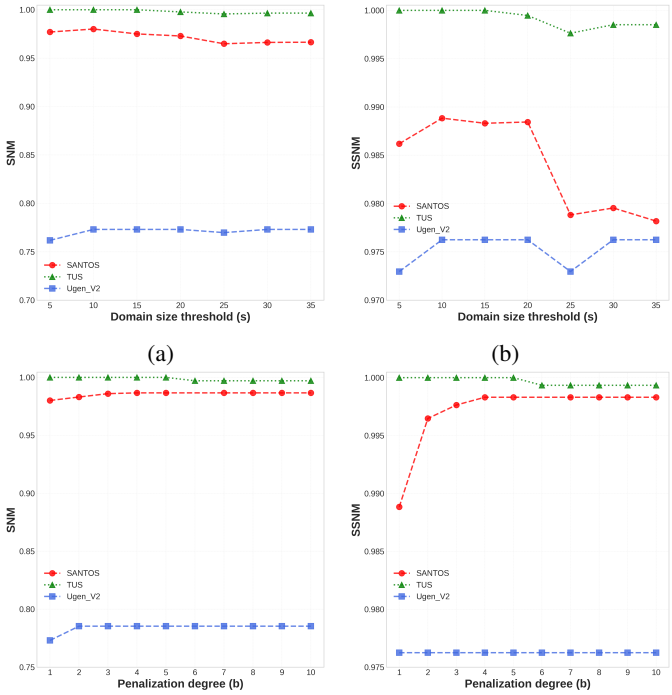


图10： b 和 s 的不同取值对 ANTs 有效性的影响，以 SNM 和 SSNM 衡量。所有数值均对所有查询和所有 $l \in [2, 10]$取平均。

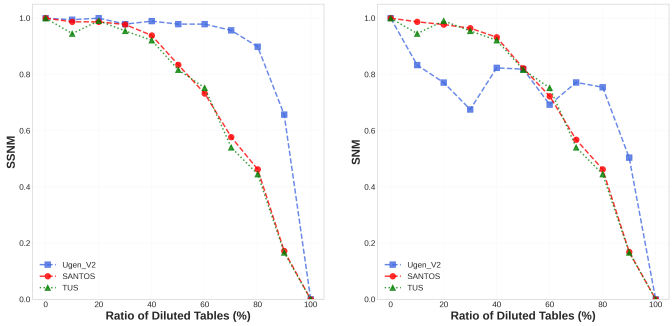


图11：初始搜索结果的冗余对 ANTs 性能的影响，按各数据集中的所有查询取平均。

随后，我们将这些合成结果输入 ANTs，并评估重排序列表的新颖性。理想情况下，我们会报告重排序结果的新颖度分数（定义 8），但事实证明，计算该分数耗时过长，难以实际进行。因此，我们报告不同稀释表比例下的 SSNM 和 SNM。对于每个数据集，我们将初始随机结果的大小固定为 15 个表，并纳入该数据集中的所有查询。然后，我们要求 ANTs 对这 15 个表进行重排序，并选出 10 个表作为最终输出。如图11a所示，在所有数据集上，就 SSNM 而言，提高初始搜索结果中的稀释表比例都会明显损害 ANTs 的性能。随着初始结果集的冗余程度增加，ANTs 的有效性随之下降。这种下降起初是逐渐变化的，这表明使用 ANTs 进行重排序能够应对中等程度的冗余。然而，当稀释表的比例在 Santos 和 TUS 上超过 40%，在 Ugen-v2 上约超过 70% 时，性能下降会变得陡峭得多。SNM 在 Santos 和 TUS 上也呈现类似趋势（图11b）。不过，对于 Ugen-v2，数据集稀疏得多：大多数查询只有少量可联合表，且目标比例并非总能达到。这解释了在 40% 和 70% 比例附近观察到的异常增长。