arXiv:2601.15758v1 [cs.DB] 22 Jan 2026

NL4ST：面向时空数据库的自然语言查询工具

Xieyang Wang、Mengyi Liu、Weijia Yi、Jianqiu Xu*南京航空航天大学，南京，中国
{xieyang,liumengyi,wjyi_x,jianqiu}@nuaa.edu.cn

Raymond Chi-Wing Wong 香港科技大学 中国香港 raywong@cse.ust.hk

摘要

移动计算设备和定位技术的发展，推动了数据库中管理的时空数据呈爆炸式增长。此类数据的典型查询包括范围查询、最近邻查询和连接查询。然而，编写此类查询通常需要领域专业知识并熟悉可执行查询语言，这对非专业用户来说是一项艰巨的任务。因此，时空数据库对易于使用的自然语言查询（NLQ）有着极大的需求。为弥合非专业用户与数据库查询计划之间的差距，我们推出了*NL4ST*，这是一款允许用户使用自然语言查询时空数据库的交互式工具。NL4ST采用三层架构：(i) 用于知识准备的知识库和语料库，(ii) 用于实体链接的自然语言理解，以及(iii) 生成物理计划。我们的演示将展示NL4ST如何生成有效的时空物理计划，并通过四个真实和合成数据集进行验证。我们已将NL4ST上线 [1]，并在 https://youtu.be/-J1R7R5WoqQ 提供演示视频。

ACM 参考文献格式：
Xieyang Wang、Mengyi Liu、Weijia Yi、Jianqiu Xu 和 Raymond Chi-Wing Wong。2026 年。NL4ST：面向时空数据库的自然语言查询工具。载于 《论文集》。 ACM，美国纽约州纽约市，5 页。https://doi.org/XXXXXXX.XXXXXXX

1 引言

在过去几十年中，尽管时空数据库已在广泛的应用中得到深入研究[6]，，但假设用户能够高效地构造准确查询，这对不熟悉时空知识和结构化数据库查询语言的非专业用户而言构成了挑战

为此，人们 increasingly 重视为数据库提供自然语言接口（NLIDB） [3]。

在相关文献中，文本到 SQL [1, 4, 8] 领域已取得显著进展。然而，这些方法主要关注关系型数据库，并将 SQL 视为最终的语义表示。

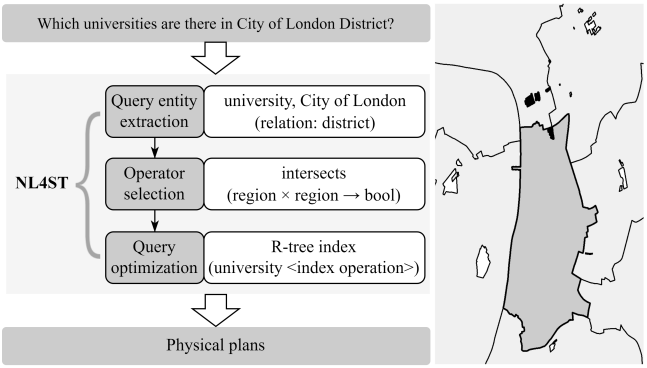

图1：时空数据库中自然语言查询转换示例。

从语义角度来看，SQL 缺乏原生的时空运算符，而且自然语言查询（NLQ）往往会导致映射结果存在歧义。例如，在图1中，短语“在伦敦市区”可能会映射为 SQL 中不同的空间谓词，从而产生“*contains (tuple.location)*”或“*inside (tuple.location)*”这类含义不明确的结构。从执行角度来看，SQL 是声明式语言，会隐藏执行决策，因为运算符选择和索引使用都交由优化器决定。因此，SQL 无法明确表达查询的执行意图。

在本演示中，我们研究面向时空数据库的**文本到查询计划**，它可将自然语言查询直接映射为查询计划，包括逻辑计划和物理计划。逻辑计划通过代数算子描述特定领域的查询语义，而物理计划则指定特定领域的执行算子和索引使用。我们以物理计划作为输出，使文本到查询计划的转换能够直接捕捉自然语言查询的语义，并明确控制执行语义和效率。

尽管查询计划提供了一种细致的数据操作方式，但文本到查询计划面临以下三项挑战。**挑战 1：将文本到 SQL 方法应用于文本到查询计划的难点**。现有文本到 SQL 方法依赖大语言模型（LLM），而时空查询计划需要丰富的领域知识和复杂的可执行结构，使训练数据构建成本高昂。此外，当前的效率控制技术通常假设系统处于黑盒环境，从而在 (i) 领域知识对齐和 (ii) 正确且高效的计划生成方面带来挑战。

[1] https://nl4st.cpolar.top/nl2secondo/
*通讯作者

允许免费制作本作品的全部或部分内容的数字或纸质副本，供个人或课堂使用，但前提是不得为营利或商业利益制作或分发副本，且副本首页须附上此声明和完整引文。作者以外的其他权利人拥有的作品组成部分，其版权必须受到尊重。注明出处后可以摘录。若要以其他方式复制、重新出版、发布到服务器或分发给列表，则须事先取得明确许可和/或支付费用。请通过 permissions@acm.org 申请许可。，
© 2026 版权归所有者/作者所有。出版权已授权给 ACM。
https://doi.org/XXXXXXX.XXXXXXX

挑战 2：缺乏领域专属知识。基于大语言模型的系统缺乏领域专属知识，导致产生幻觉和实体对齐错误。如图1所示，GPT-4o 常常在列级别上错误对齐“伦敦市区”，尽管该实体对应的是特定的值级表示，这凸显了采用领域感知的实体落地来增强自然语言理解的必要性。

挑战3：查询计划生成的复杂性。算子和索引的巨大组合空间，使得物理计划生成难以在保持高效的同时实现扩展。对于图1中的自然语言查询，GPT-4o 使用了*ininterior* 算子，而不是 *intersect* 算子，因此未能利用可用的索引。

关于时空领域中 NLIDB 的研究十分有限 [5, 7], ，因为将空间和时间维度整合到统一框架中会带来复杂性。

我们开发了 NL4ST，这是一种面向时空数据库的自然语言查询工具，使用户能够使用自然语言查询空间数据、时间数据和轨迹数据。与文本到SQL相比，文本到查询计划的方法（不涉及中间结果SQL）能够：(i) 避免文本语言中的歧义，以及 (ii) 通过引入特定领域的算子和数据类型来增强表达能力。该工具由三个层次组成：(i) 知识库和语料库，(ii) 自然语言理解，以及 (iii) 物理计划生成。NL4ST 通过数据库分析生成知识库，并通过收集顶级会议和期刊已发表论文中的自然语言查询来构建语料库，随后利用大语言模型对其进行扩充。知识库和语料库被用于实现知识检索和示例参考，指导用户提交稳健的查询；同时，它们也可以独立用于后续的自然语言理解。NL4ST 使用自然语言处理（NLP）工具进行粗粒度实体抽取，随后结合高级算法和知识库进行细粒度实体抽取。NL4ST 利用查询类型分类模型以及与查询类型相对应的结构化语言模型，从而为物理计划生成剪枝候选集。此外，NL4ST 还采用模型预测最优物理计划。该系统已在线发布，可通过 https://nl4st.cpolar.top/nl2secondo/ 访问。

2 系统架构

我们在图2中给出了 NL4ST 的框架，该框架采用三层架构：（i）知识准备、（ii）自然语言理解和（iii）物理计划生成。

2.1 知识准备

NL4ST 通过（i）知识库生成器和（ii）语料库生成器解决时空领域知识不足的问题。

知识库生成器。 我们构建了一个时空知识库，用于支持知识检索和实体候选项剪枝。该知识库由关系知识库和位置知识库组成。关系知识库存储包含标识符、名称和时空属性的关系表，而位置知识库则包含以点、线和区域 表示的对象。这两个知识库

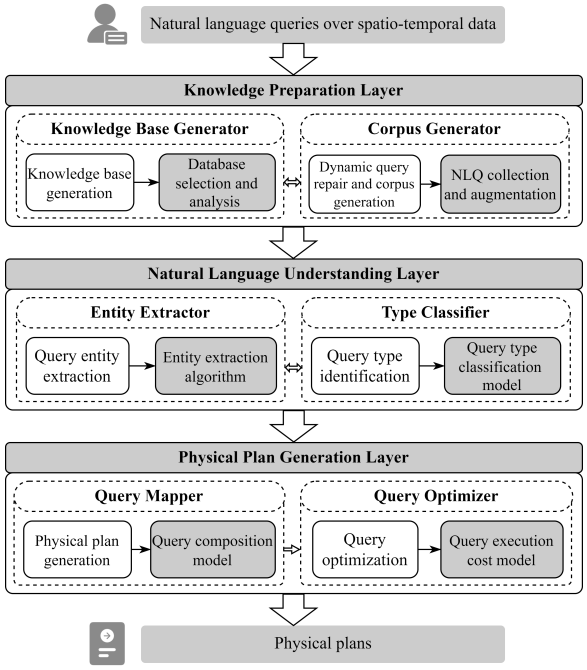

图2：NL4ST 的架构概览。

紧密耦合，其中位置条目引用关系知识库中定义的标识符。

语料库生成器。 我们构建了一个包含约 5,000 个模板的自然语言查询语料库，涵盖空间查询和移动对象查询，以支持用户引导以及后续的查询类型识别。自然语言查询取自已发表论文，并通过带有预设提示词的 GPT-4o 进行扩充。为确保与数据库保持一致，我们采用自动检测与修复方法对自然语言查询进行完善。我们还构建了模板，以便针对不同数据库生成大量高质量的自然语言查询。该语料库由查询类型和自然语言查询组成。自然语言查询中的关系和实体均来源于所选数据库。

2.2 自然语言理解

NL4ST 主要执行自然语言处理，包括两个模块：(i) 实体提取器和 (ii) 类型分类器。

实体提取器。 该提取器将自然语言查询中的实体与数据库值对齐。*spaCy* 用于初始实体识别，它是一种自然语言处理工具；随后进行候选实体生成，得到两个列表：(i) 数值列表，包含标记为时间、数字、基数或数量的实体；(ii) 信息列表，可能包含关系、位置或对象。然后，我们通过细粒度提取算法对候选项进行剪枝，以提取确切的关键实体。对象标识符和最近邻标识符涵盖与对象及“最近”等关键词相关的上下文信息。借助短语中的数值及其关联的距离单位，可以确定精确的距离阈值。除此之外，我们利用构建的时空知识库，对信息列表中的关系、对象和位置信息进行直接检索。

类型分类器。 鉴于时空查询的多样性，为所有查询类型提供一个通用的翻译模型并不现实。识别并分类查询类型，可以显著缩小构建物理计划时的搜索空间。LSTM具有良好的性能且训练时间较短，因此，我们利用给定的语料库训练LSTM模型。训练好的模型可用于类型识别。

2.3 物理计划生成

NL4ST 接下来进行物理计划生成，其中包括两个关键模块：(i) 查询映射器和 (ii) 查询优化器。

查询映射器。我们基于预定义的查询模型，通过映射规则组合不同查询类型所需的实体和算子，从而构建候选物理计划。我们以受空间限制的移动对象最近邻查询为例。算子 *knearest* 生成一个元组流，其结构与输入流相同。受最近的 k 个移动对象到达时间的限制，我们创建一种移动列车版本，其中各单元按开始时间排序。我们可以使用关系 *UTOrdered* 来获取物理计划。查询表达式的模板如下，其中“”内的内容将从自然语言查询中提取。

查询 *UTOrdered feed filter [(deftime(.UTrip)* ***intersect***算子〈*period*〉)*] **knearest**[UTrip,*〈 *object* 〉,〈 *k* 〉]*consume;*

查询优化器。 为了进一步提升易用性，我们通过关注索引利用来优化查询效率。我们统计并收集每个时空关系中涉及的记录数。然后，根据算子约束以及查询中涉及的关系估算过滤率。过滤率是指数据集中符合特定查询条件的记录数占总记录数的比例。对于过滤率较高的查询，我们为查询映射器生成的查询枚举带索引的候选物理计划。随后，我们采用代价模型，使用抽样小规模数据执行这些物理计划，以预测每个候选物理计划的时间开销，最终为用户提供最优物理计划。

3 演示

我们在一台运行 Ubuntu 22.04（64 位，内核版本 6.8.0-49-generic）的台式电脑上部署 NL4ST（Intel(R) Core(TM) i9-11950H CPU，2.60 GHz，32 GB 内存，512 GB 磁盘）。我们为用户提供交互式网页用户界面，并将后端集成到可扩展数据库系统 SECONDO [2]中，作为一个代数模块。用户可根据已加载的数据集执行查询，并获得物理计划和可视化结果。我们使用了四个数据集，其中三个涉及城市数据[2]，第四个则聚焦于水文数据[3]。(i) 数据集 *nanjingtest* 包含南京的出租车、区、部分道路和兴趣点。(ii) 数据集 *londontest* 包含伦敦部分兴趣点和区。(iii) 数据集*berlintest* 包含柏林的城市数据，包括列车、兴趣点和河流。(iv) 数据集*chinawater* 存储了中国的部分水系。详细统计数据见表 1。我们展示 NL4ST 将时空自然语言查询转换为物理计划的能力，旨在为用户提供交互式工具，帮助他们了解上述文本到查询计划技术的工作机制。

表 1：数据统计。

数据集	#表	#点	#线	#区域	#移动对象
nanjingtest	6	9,000	887	13	147
londontest	6	9,032	9,728	12,669	0
berlintest	50	3,040	4,078	330	562
chinawater	2	0	8,399	2,907	0

3.1 演示场景

图3展示了 NL4ST 的截图，用户可以使用它执行 (i) 基本空间查询、(ii) 时间区间查询、(iii) 范围查询、(iv) 最近邻查询、(v) 连接查询、(vi) 相似性查询和 (vii) 聚合查询。我们将按以下步骤进行演示，并提供两个查询场景。

语料库生成和示例参考。 首先，用户可以选择不同数量的语料库数据进行生成，并利用生成的时空查询语料库训练自己的模型。语料库中的这些自然语言查询示例也可以指导用户构造高质量的查询。

知识检索。 随后，用户可以从预先构建的知识库中检索实体，并利用这些实体结合示例来构造后续的自然语言查询。

数据库选择和自然语言查询输入。 由于模式会影响预期物理计划的执行，用户必须先为查询选择目标数据库。为简化这一步，用户可以查看示例查询，并直接选择它们作为输入。输入自然语言查询后，用户可以点击“提交”按钮。

执行结果显示。 提交后，工具会展示解析过程。首先进行粗粒度实体识别，随后进行细粒度实体消歧，再执行详细的实体抽取，以便进行查询模板实例化。系统根据抽取的信息生成物理计划，并返回相应的查询结果。**地图可视化与交互**。 用户可以点击“在 SECONDO 中显示”按钮，在数据库中查看可视化结果。该工具还会在交互式地图上展示结果。对于每种查询类型，用户都可以获得量身定制的可视化设计，这也有助于用户发现结果在现实世界中的用例。

错误处理。 当输入包含不支持的查询类型、语法错误或实体相关错误时，用户可以点击“帮助”按钮。工具会解释错误原因并提供可能的修正方案。用户可以根据建议修改输入，然后重新提交查询。

查询效率显示。最后，该工具会报告物理计划的执行时间。用户可以点击“切换”按钮，对比原始查询与优化查询的执行时间。用户也可以点击“算子树”按钮，通过可视化算子树查看物理计划。

表2：示例查询。

自然语言查询示例	Type
Q1：伦敦每所大学附近有哪些快餐店?	空间连接查询
Q2：请显示早上 6 点到 11 点之间，距离 5 号列车最近的 50 个邻居。	移动对象最近邻查询

[2]https://github.com/zhongmove/NL4ST [3] https://www.hydrosheds.org/products

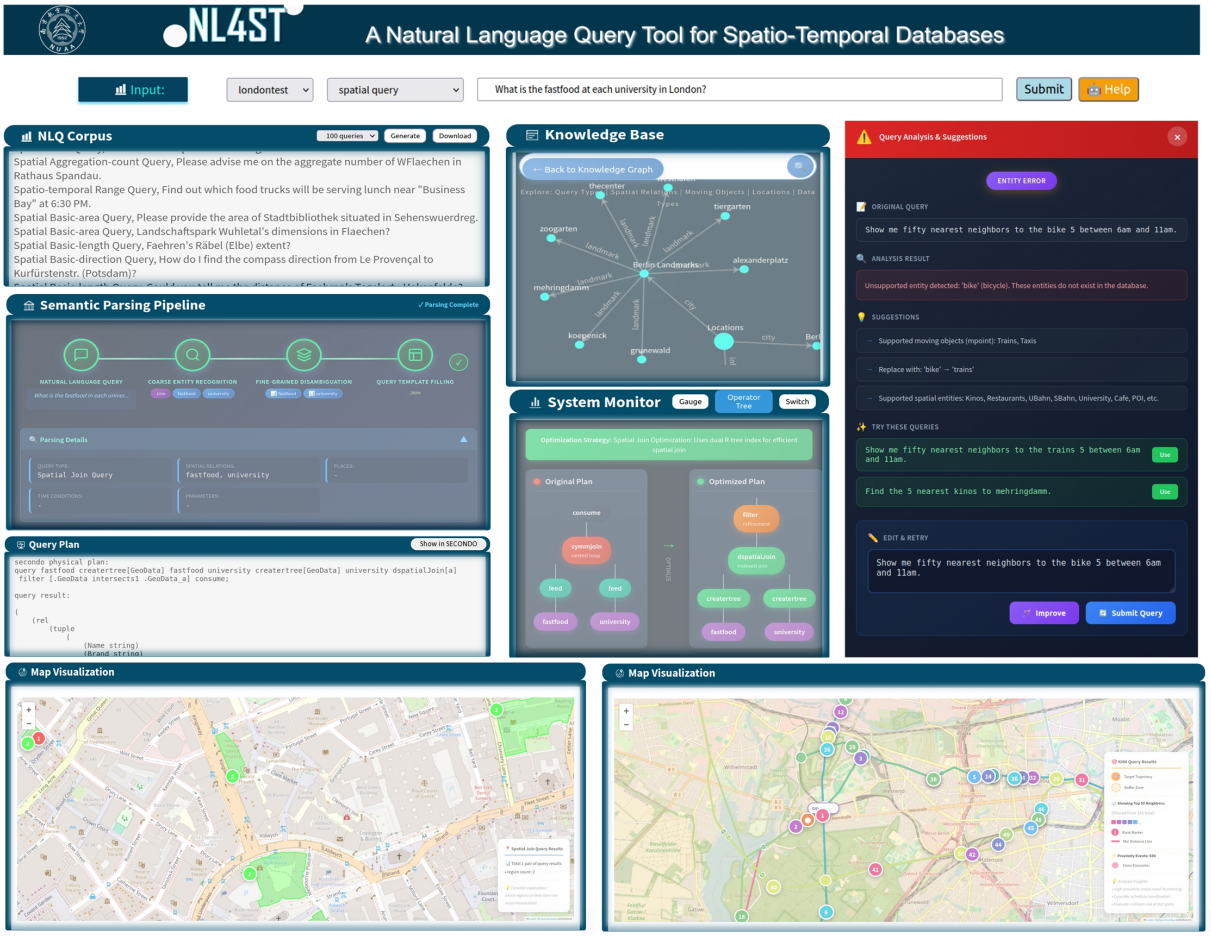

图3：NL4ST 的截图。

场景1：查询空间数据。我们以表2中给出的空间连接查询 *Q1* 为例，按照上述步骤演示系统的工作流程。用户首先可以查看语料库和知识库，其中提供了一组标准查询和实体以供参考。随后，用户选择目标数据库并提交自然语言查询，也可以选择系统推荐的查询。图3展示了详细的语义解析过程；系统据此生成物理计划，并在地图上可视化查询结果。NL4ST 使用 R树索引进行优化，用户可以查看系统监视器以获取查询执行时间。用户还可以通过可视化算子树查看查询执行过程，点击“算子树”按钮即可访问。

场景2：查询移动对象。表2所示的最近邻查询*Q2*同样遵循上述步骤生成结果并提供可视化。在地图可视化模块中，NL4ST 使用连接线表示最近邻顺序和距离。该可视化可引导用户发现现实世界中的应用场景，例如识别移动对象彼此接近的位置或时间段。如果存在问题，例如图3所示的实体相关错误，用户可以点击“帮助”按钮获取错误解释和指导。在分析结果区域中，用户可以查看详细的错误消息，而建议区域则提供潜在修正。用户还可以直接尝试列出的推荐查询。

3.2 评估指标

我们使用三项指标评估 NL4ST：(i) 响应时间、(ii) 可翻译性和 (iii) 翻译精确率。可翻译性衡量能够转换为物理计划的自然语言查询所占比例，而翻译精确率衡量转换后的物理计划中能够产生预期结果的比例。NL4ST 的平均响应时间、可翻译性和翻译精确率分别为 1.9 秒、93% 和 90%。

参考文献

[1] Jiaqi Guo、Zecheng Zhan、Yan Gao 等。2019。面向跨领域数据库中具有中间表示的复杂文本到SQL。在*ACL*。4524–4535。[2] Ralf Hartmut Güting、Thomas Behr 和 Christian Düntgen。2010。SECONDO：用于移动对象数据库研究以及研究实现的发布与集成的平台。*IEEE Data Eng. Bull.* 33，2（2010），56–63。[3] George Katsogiannis-Meimarakis、Mike Xydas 和 Georgia Koutrika。2023。基于深度学习的数据库自然语言接口。*Proc. VLDB Endow.* 16，12（2023），3878–3881。[4] Fei Li 和 H. V. Jagadish。2014。为关系数据库构建交互式自然语言接口。*Proc. VLDB Endow.* 8，1（2014），73–84。

[5] Mengyi Liu、Xieyang Wang、Jianqiu Xu 等。2025。NALSpatial：空间数据库的自然语言接口。*IEEE Transactions on Knowledge and Data Engineering*37，4（2025），2056–2070。[6] Yongxin Tong、Jieying She、Bolin Ding 等。2016。空间众包中的在线移动微任务分配。在*ICDE*。49–60。

[7] Xieyang Wang、Mengyi Liu、Jianqiu Xu 等。2023。NALMO：为移动对象数据库转换自然语言查询。*GeoInformatica* 27, 3 2023 427 460 ( ), – . [8] Xiangjin Xie、Guangwei Xu、Lingyan Zhao 等。2025。OpenSearch-SQL：通过动态少样本和一致性对齐增强文本到SQL。*ACM*管理数据会议录 3, 3 (2025), 194:1–194:24。