

# Monkuu: 一个基于 LLM 和动态模式映射的地理空间数据库自然语言接口

原标题: Monkuu: a LLM-powered natural language interface for geospatial databases with dynamic schema mapping

期刊: International Journal of Geographical Information Science (IJGIS)

发表时间: 2025年7月17日

作者: Chenglong Yu, Yao Yao (通讯作者), Xiang Zhang 等

单位: 中国地质大学(武汉)、LocationMind Inc.、东京大学等

## 摘要 (Abstract)

地理空间数据库由于结构化查询语言(SQL)的复杂性,一直存在巨大的访问门槛。为了通过自然语言实现直观的人机交互,本文提出了 **Monkuu**——一种专为地理空间数据库设计的新型“自然语言转 SQL”(NL2SQL)接口。

Monkuu 集成了一种动态上下文感知模式映射(**Dynamic Context-aware Schema Mapping**)机制来对齐数据库模式,有效地克服了传统“检索增强生成”(**RAG**)方法中常见的信息截断问题。此外,我们引入了一个“人在回路”(**Human-in-the-loop**)的地理消歧工作流,通过结合多源地理数据来解决复杂地名的问题。

Monkuu 在 KaggleDBQA 基准测试中达到了 **56.2%** 的执行准确率,比领先的 ZeroNL2SQL 模型提高了 **13.8** 个百分点;同时在 GeoQueryJP 数据集上实现了 **82.4%** 的地理歧义解决召回率。该系统的主要贡献在于其强大的数据库访问能力,为下游空间分析工具提供了干净的数据接口,同时保持了查询转换的高准确性。案例研究展示了其处理诸如“显示柏市(Kashiwa)的边界”等查询并将其转化为可执行 SQL 的有效性,显著降低了非专家用户的技术门槛。这项工作推动了地理信息服务的公平性和可访问性。

关键词: 大语言模型;数据库接口;模式映射(Schema mapping);NL2SQL;地理空间实体消歧

## 1. 引言 (Introduction)

GIS 是智慧城市和空间决策的基石技术,但在创新交互范式以实现民主化方面仍面临挑战。大语言模型(LLM)的出现正在从根本上重塑人工智能。从 BERT 到 GPT-4, LLM 在跨领域知识整合和复杂推理方面展现了惊人的能力,正成为化学、法律和金融等领域强大的问题解决引擎。

这种智能转型也影响了 GIS 研究。然而,目前存在一个关键瓶颈:如何通过自然语言无缝、直观地访问存储在结构化地理空间数据库中的海量信息?现有系统往往难以应对地理空间数据和查询语言的固有复杂性,且依赖固定文件路径的数据输入方式阻碍了动态交互。

通过自然语言访问地理空间数据库面临独特的挑战:

1. **NL2SQL** 的通用难题: 如语义解析和模式匹配(Schema Matching)。
2. **GIS** 的特殊复杂性: 涉及复杂的空间数据类型、关系和算子(如缓冲区分析、空间连接)。

### 3. 地名歧义 (Geographic Name Ambiguity)

**Ambiguity**：地名可能指代多个位置，严重增加了查询解释的难度。

为了解决这些挑战，本文介绍了 **Monkuu**。它通过两个核心创新来降低非专家用户的技术门槛：

1. 动态上下文感知模式映射机制：精确对齐自然语言查询与复杂的数据库模式，专门用于缓解传统 RAG 方法中的信息截断问题。
2. 人在回路的地理消歧工作流：结合多源地理数据和人工监督，准确解决用户查询中模糊或复杂的地名。

Monkuu 的主要贡献在于增强了将自然语言转化为可执行地理空间 SQL 的准确性和鲁棒性。

## 2. 相关工作 (Related Work)

### 2.1 用于地理空间分析和任务自动化的 LLM 驱动 Agent

这一类研究利用 LLM 自动化复杂的 GIS 工作流。例如 LLM-Geo 和 GIS Copilot，它们能够将空间问题分解为多步过程，生成 Python 代码并编排 GIS 工具（如 QGIS）。这类研究通常侧重于分析过程的执行，而较少关注如何通过自然语言从复杂的结构化数据库中提取特定数据子集。

### 2.2 用于广泛地理空间数据检索和专门任务的 LLM 系统

另一类研究关注利用 LLM 发现和检索数据，如 LLM-Find，或用于特定任务（如地址标准化）的 GeoAgent。这些系统擅长获取整个数据文件或回答事实性问题，但通常不专注于生成用于深度交互关系型地理空间数据库的结构化查询语言（SQL）。

### 2.3 地理空间环境下的数据库自然语言接口 (NLIDB)

NLIDB 旨在简化 SQL 使用。在地理空间领域，这一挑战被放大，因为涉及特殊的空间结构和操作。现有的努力（如 NALSpatial）通过将自然语言转换为自定义的“可执行语言”取得了进展，但依赖专有的中间语言可能会限制互操作性。因此，生成标准 SQL、鲁棒地处理复杂的地理空间模式并有效解决地名歧义，仍然是一个重大的开放性挑战。

Monkuu 的填补空白：

Monkuu 旨在通过生成标准 SQL 来增强互操作性；通过动态模式映射解决复杂模式对齐；通过“人在回路”的工作流解决地名歧义。

## 3. 方法论 (Methodology)

### 3.1 系统整体架构

Monkuu 的架构（见原文图1）包括数据处理工作流、中间件、NL2SQL 接口和地理空间处理模块。

- 数据层：对原始数据建立空间索引(如 R-tree)并存储在空间数据库中。关键步骤是专家验证并生成标准化模式文档(**Schema Document**)，这为下游组件提供了重要的 GIS 上下文信息。
- 中间件层：管理 SQL 执行器(包含权限控制)并与\*\*模式映射器(Schema Mapper)\*\*协作。
- **NL2SQL** 接口层：执行意图识别、地理消歧、模式获取和 SQL 生成。

形式化地, NL2SQL 任务表示为生成可执行 SQL 查询 \$Y\$:

\$\$Y = f(Q, S, I | \theta)\$\$

其中 \$Q\$ 是自然语言问题, \$S\$ 是数据库模式信息(表、列、外部知识), \$I\$ 是系统指令, \$f\$ 是参数为 \$\theta\$ 的 LLM。

## 3.2 具有地理空间提示工程的核心 NL2SQL 管道

核心流程包含三个关键模块:查询分类器、地理实体提取器和模式映射器, 最后由 **SQL** 生成器输出查询。

### 3.2.1 用户意图识别 (Query Classifier)

作为入口, 该模块将用户查询分类为预定义的类别:

- Geo\_Query\_Can\_Solve: 可处理的地理查询(如“显示大宫站”)。
  - Chat\_Query: 闲聊(如“你有多少数据?”)。
  - Geo\_Query\_Cannot\_Solve: 超出范围的查询(如数据覆盖范围外)。
- 这通过结构化的 Prompt 实现, 包含任务指令和示例。

### 3.2.2 地理实体提取 (GeoEntity Extractor)

这是消歧工作流的关键。它从用户查询中识别并提取地名及其可能的类型(如 Point 点或 Area 面)。

- 例如: 从>Show me the Omiya Station”中提取 {"entity": "Omiya Station", "type": "Point"}。

### 3.2.3 地理空间数据库的模式映射 (Schema Mapper)

\$\$核心创新\$\$

为模型提供足够的上下文信息至关重要。Monkuu 提出了 **Schema Mapper** 工作流。

- 挑战: LLM 的上下文窗口有限(通常约 32K token), 不能塞入所有数据库信息。
  - 解决方案: 引入“模式摘要”(Schema Abstract, \$D\_a\$)。这是一个简化但完整的数据库描述。
  - 过程: \$\$T\_{relevant} = LLM(Q, D\_a, I)\$\$
- 将用户查询 \$Q\$、数据库摘要 \$D\_a\$ 和指令 \$I\$ 输入 LLM, LLM 输出与查询最相关的表名列表。这比传统的 RAG 检索更准确, 避免了关键字段丢失。

### 3.2.4 GIS 感知的 SQL 生成 (SQL Generator)

最后, 采用 Few-shot(少样本)学习方法生成 SQL。

Prompt 结构为: \$P\_0 = I \oplus S\_{related} \oplus Q\$

- \$I\$: 指令(明确使用 PostGIS 方言, 如 ST\_DWithin, ST\_Intersects)。

- $\mathcal{S}_{\text{related}}$ : 由 Schema Mapper 筛选出的相关模式。
- Few-shot Examples: 展示常见的地理空间查询模式及其对应的正确 SQL。

### 3.3 地理实体的交互式消歧 (Interactive Disambiguation)

Monkuu 通过“人在回路”(Human-in-the-loop)机制解决地名的一词多义问题(见原文图 2E)。

当存在潜在歧义实体  $E$  时, 系统启动交互流程:

#### 3.3.1 基于点的消歧 (Point-based)

对于点状位置(如具体地址或 POI), 系统利用 Google Maps Autocomplete API。

$\mathbb{E}_p^* = \sigma(G(Q_p))$

系统列出 API 返回的候选地点(含坐标), 用户选择( $\sigma$ )匹配其意图的那一个。

#### 3.3.2 基于面的消歧 (Area-based)

对于区域查询(行政区划), 使用自定义的分层地名查询树和区域数据库(基于官方边界数据, 如日本的 Chome 级别数据)。

$\begin{cases} D_j = \sigma(T(Q_a)) \\ E_r^* = \mathcal{M}^{-1}(D_j) \end{cases}$

用户从候选区域列表中选择目标, 系统通过逆映射函数获取该区域的精确几何边界。

## 4. 实验设计与结果 (Experiments & Results)

### 4.1 评估数据集

1. **KaggleDBQA**: 一个跨领域的 NL2SQL 基准数据集, 包含复杂的查询和真实的 Web 数据库。用于评估 Monkuu 的常规 NL2SQL 性能。
2. **GeoQueryJP**: 本研究构建的专用数据集, 包含 53 个测试实例, 涵盖 9 种典型的地理歧义类型(如“同名异地”、“同一地点的不同称呼”), 用于测试地理数据查询准确性。

### 4.2 评估指标

1. 执行准确率 (**EX**): 衡量生成的 SQL 执行结果与 Ground Truth 结果是否一致。
2. 地理歧义解决召回率 (**GARR**): 衡量系统生成的候选项中是否包含用户真正想要查询的地点。

### 4.3 结果分析

#### 模块准确性与对比

- 地理消歧效果: 在 GeoQueryJP 数据集上, 未启用消歧模块时 GARR 仅为 **40.3%**; 启用 Monkuu 的消歧模块后, GARR 提升至 **82.4%**。
- 消融研究 (**Ablation Study**): 在 KaggleDBQA 上:
  - 完整 Monkuu 系统(使用 Schema Mapper): EX **56.2%**
  - 使用 RAG 替代 Schema Mapper: EX **50.3%**
  - 直接使用全量模式 (Full Schema): EX **16.2%**(证明了直接把所有表结构塞给 LLM 会导致严重的干扰)。

## 与其他模型对比

Monkuu (56.2%) 显著优于传统模型 EditSQL (11.7%)、RAT-SQL (13.6%)、RESDSQL (31.9%) 以及 ZeroNL2SQL (42.4%)。

## 4.4 案例研究 (Case Study)

1. 地理边界检索：
  - 用户：“Show me Kashiwa City boundary.(给我看柏市的边界)”
  - Monkuu：识别意图 -> 映射到 boundary\_city 表 -> 生成 SQL SELECT \* FROM boundary\_city WHERE city\_name = 'Kashiwa' -> 在地图上渲染粉色多边形。
2. 带有多重约束的 POI 查询：
  - 用户：“At convenience stores in Shinjuku Ward, which are the top 10 stores that have been visited the most in the past month?(在新宿区的便利店中, 过去一个月访问量最大的前10家是哪些?)”
  - Monkuu：
    1. 筛选地点:新宿区 (Shinjuku Ward)
    2. 筛选行业:便利店 (Convenience stores)
    3. 语义映射：将“visited the most in the past month”智能映射到数据库字段 population\_month\_latest。
    4. 生成包含 ORDER BY 和 LIMIT 10 的复杂 SQL。

## 5. 讨论 (Discussion)

### 5.1 进展

Monkuu 的 **Schema Mapper** 是其核心优势。实验证明, 相比于 RAG(容易截断信息)和 Full-Schema(信息过载), Schema Mapper 通过动态筛选最相关的表结构, 为 LLM 提供了最佳的上下文环境。同时, 生成标准 SQL 保证了其能与现有的 GIS 工具链无缝集成。

### 5.2 局限性

- 依赖文档质量：Schema Mapper 的准确性高度依赖于模式文档(Schema Document)的质量和详细程度。如果文档描述不清, 映射就会出错。
- 消歧的自动化程度：目前仍需“人在回路”, 虽然准确但影响了自动化效率。
- 地理覆盖范围：目前的区域数据库主要覆盖日本, 限制了全球适用性。

### 5.3 未来工作

- 通过集成全球行政边界数据库(如 GADM)和地理编码服务来扩展消歧范围。
- 研究如何处理用户上传的数据集(交互式模式定义)。
- 探索采用标准通信协议(如 Model Context Protocol, MCP)以增强互操作性。

## 6. 结论 (Conclusion)

Monkuu 通过引入交互式消歧工作流和动态模式映射策略, 有效地解决了地理空间数据库访问中的关键难题。它不仅在基准测试中取得了最先进的性能(SOTA), 还为非专家用户提供了一个干

净、可靠的数据接口，让地理空间数据变得更加触手可及。