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

Chenglong Yu, Yao Yao, Xiang Zhang, Geyuan Zhu, Yanduo Guo, Xiaowei Shao, Mariko Shibasaki, Zhihui Hu, Liangyang Dai, Qingfeng Guan & Ryosuke Shibasaki

To cite this article: Chenglong Yu, Yao Yao, Xiang Zhang, Geyuan Zhu, Yanduo Guo, Xiaowei Shao, Mariko Shibasaki, Zhihui Hu, Liangyang Dai, Qingfeng Guan & Ryosuke Shibasaki (17 Jul 2025): Monkuu: a LLM-powered natural language interface for geospatial databases with dynamic schema mapping, International Journal of Geographical Information Science, DOI: 10.1080/13658816.2025.2533322

To link to this article: https://doi.org/10.1080/13658816.2025.2533322

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
https://doi.org/10.1080/13658816.2025.2533322

RESEARCH ARTICLE

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

Chenglong Yu[a,b], Yao Yao[a,b,c,d], Xiang Zhang[a,b], Geyuan Zhu[a,b], Yanduo Guo[a,b,e], Xiaowei Shao[b,d], Mariko Shibasaki[b], Zhihui Hu[a], Liangyang Dai[a], Qingfeng Guan[a] and Ryosuke Shibasaki[b,d,f]

[a]UrbanComp Lab, School of Geography and Information Engineering, China University of Geosciences. Wuhan, Hubei, China; [b]LocationMind Institution, LocationMind Inc, Chiyoda, Tokyo, Japan; [c]Hitotsubashi Institute for Advanced Study, Hitotsubashi University, Kunitachi, Tokyo, Japan; [d]Faculty of Engineering, Reitaku University, Kashiwa, Chiba, Japan; [e]School of Computer Science, China University of Geosciences, Wuhan, China; [f]Interfaculty Initiative in Information Studies and Graduate School of Interdisciplinary Information Studies, The University of Tokyo, Tokyo, Japan

ABSTRACT

Geospatial databases present significant accessibility challenges due to the complexity of structured query languages. To enable intuitive human-system interactions via natural language, this paper presents Monkuu, a novel natural language-to-SQL interface specifically designed for geospatial databases. Monkuu integrates a dynamic context-aware schema mapping mechanism to align database schemas, effectively overcoming information truncation issues common in traditional Retrieval-Augmented Generation methods. Additionally, a human-in-the-loop geographic disambiguation workflow is introduced to resolve complex place names by combining multi-source geographic data. Monkuu achieves 56.2% execution accuracy on the KaggleDBQA benchmark, improving upon the leading ZeroNL2SQL model by 13.8 percentage points, alongside an 82.4% recall in geographic ambiguity resolution on the GeoQueryJP dataset. The system's primary contribution lies in its robust database access capabilities with clean data interfaces for downstream spatial analysis tools while maintaining focus on accurate query translation. Case studies demonstrate its effectiveness in processing queries like 'Show me the boundary of Kashiwa' into executable SQL, significantly lowering technical barriers for non-expert users. This work advances equitable and accessible geographic information services.

HIGHLIGHTS

1. Converts natural language to geospatial SQL via dynamic schema mapping.
2. 56.2% accuracy on KaggleDBQA (+13.8% over ZeroNL2SQL)
3. Hybrid disambiguation boosts location resolution accuracy to 82.4% on GeoQueryJP.

ARTICLE HISTORY
Received 14 March 2025
Accepted 9 July 2025

KEYWORDS
Large language model; database interface; schema mapping; nl2sql; geospatial entity disambiguation

CONTACT Yao Yao yaoy@cug.edu.cn, ya_yao@locationmind.com, yaoy@csis.u-tokyo.ac.jp

Supplemental data for this article can be accessed online at https://doi.org/10.1080/13658816.2025.2533322.

4. Introduces Schema Mapper for enhanced SQL query generation accuracy.
5. Modular design enables downstream spatial tool integration.

1. Introduction

GIS is a cornerstone technology for smart cities and spatial decision-making, yet it continues to face challenges in innovating interaction paradigms to enhance their democratization (Nelson *et al.* 2022). The advent of Large Language Models (LLM) is fundamentally reshaping artificial intelligence. The developmental trajectory of language models traces from early statistical language models (Jelinek 1990) through neural network-based distributed word embeddings like Word2Vec (Church 2017) to the Transformer architecture (Vaswani 2017)powered pre-trained models. From BERT (Devlin *et al.* 2019) to GPT-4 (Achiam *et al.* 2023), language models have undergone dual evolutionary leaps in both parameter scale and cognitive capabilities. With their remarkable capabilities in cross-domain knowledge integration and complex reasoning, LLM is becoming powerful domain-specific problem-solving engines, as seen in fields like chemistry (Boiko *et al.* 2023), law (Yao *et al.* 2024), and finance (Wu *et al.* 2023).

This intelligent transformation is also influencing GIS research, with scholars integrating LLM to handle geospatial challenges (Hu *et al.* 2023, Zhang *et al.* 2024a, Akinboyewa *et al.* 2024). However, a critical bottleneck persists in achieving seamless and intuitive natural language access to the wealth of information stored in structured geospatial databases. Current systems often struggle with the inherent complexities of geospatial data and query languages, and limitations such as reliance on fixed file paths for data input hinder dynamic data interaction scenarios. This underscores the need for more adaptive and intelligent interfaces.

Accessing geospatial databases via natural language presents distinct challenges. Beyond the general difficulties of Natural Language to SQL (NL2SQL) tasks – such as semantic parsing and schema matching (Li and Jagadish 2014, Sutskever *et al.* 2014, Kim *et al.* 2020, Li *et al.* 2023) – geospatial databases involve intricate spatial data types, relationships, and operators (e.g. buffer analysis, spatial joins). Furthermore, the pervasive issue of geographic name ambiguity (Smith and Crane, 2001, Amitay *et al.* 2004, Garbin and Mani 2005, Goodchild *et al.* 2005), where place names can refer to multiple locations, significantly complicates query interpretation and demands robust disambiguation.

To address these challenges, this paper introduces Monkuu, a novel natural language-to-SQL interface specifically designed for geospatial databases. Monkuu aims to lower technical barriers for non-expert users and provide accurate, reliable data access for downstream spatial analysis by incorporating two core innovations. It features a dynamic context-aware schema mapping mechanism that precisely aligns natural language queries with complex database schemas, specifically designed to mitigate information truncation issues common in traditional Retrieval-Augmented Generation (RAG) methods.

Additionally, Monkuu integrates a human-in-the-loop geographic disambiguation workflow, which combines multi-source geographic data and human oversight to accurately resolve ambiguous or complex place names in user queries. The primary contribution of Monkuu lies in its enhanced accuracy and robustness in translating natural language into executable geospatial SQL. This is evidenced by its performance, achieving 56.2% execution accuracy on the KaggleDBQA benchmark—an improvement of 13.8 percentage points over the leading ZeroNL2SQL model—and an 82.4% recall in geographic ambiguity resolution on the GeoQueryJP dataset. Case studies, such as processing queries like "Show me the boundary of Kashiwa" into executable SQL, further demonstrate Monkuu's effectiveness in simplifying interaction with complex geospatial data and advancing equitable, accessible geographic information services.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 details the architecture and methodologies of Monkuu. Section 4 presents the experimental setup and results. Section 5 discusses the findings and limitations, and Section 6 concludes the paper.

2. Related work

The advent of LLM has significantly reshaped the landscape of GIS, opening new avenues for natural language interaction, spatial analysis automation, and enhanced data accessibility. This section reviews existing literature, organized by key thematic approaches, to contextualize the contributions of Monkuu. We will explore themes including: (1) LLM-driven agents for geospatial analysis and task automation; (2) LLM-powered systems for broad geospatial data retrieval and specialized task execution; and (3) Natural Language Interfaces for Databases (NLIDB), with a specific focus on the unique challenges and advancements within the geospatial domain.

2.1. LLM-driven agents for geospatial analysis and task automation

A prominent research theme involves leveraging LLM to automate complex geospatial analysis workflows and tasks, effectively creating intelligent GIS agents. These systems typically aim to interpret high-level natural language instructions from users and translate them into executable operations within GIS environments. For instance, LLM-Geo (Li and Ning 2023) and GIS Copilot (Akinboyewa *et al.* 2024) demonstrate how LLM can decompose spatial problems into multi-step procedures, generate Python code, and orchestrate GIS tools (e.g., within QGIS) to perform tasks ranging from data visualization to complex spatial modeling. Other notable examples include MapGPT (Zhang *et al.* 2024b), GeoGPT (Zhang *et al.* 2024a). The primary focus of these agents is often on the execution of analytical processes and the generation of analytical outputs. While demonstrating impressive capabilities in task automation, this line of research generally assumes that the necessary data is already prepared and accessible, with less emphasis on the nuanced challenges of extracting specific data subsets from complex, structured geospatial databases via natural language.

2.2. LLM-powered systems for broad geospatial data retrieval and specialized tasks

Another thematic area concerns the use of LLM for discovering and retrieving geospatial data from diverse sources, or for executing highly specialized geospatial tasks. Systems like LLM-Find (Ning *et al.* 2025) aim to create GIS agent frameworks that can autonomously find and retrieve geospatial datasets from online repositories or local storage based on natural language descriptions, often by generating and debugging data acquisition programs. Question-answering agents like Geode (Gupta *et al.* 2024) leverage LLM to parse geospatial queries and retrieve factual information from various sources, including APIs and knowledge bases, to provide direct answers. Furthermore, specialized agents like GeoAgent (Huang *et al.* 2024) focus on niche problems such as address standardization by integrating LLM with specific geospatial tools and knowledge. These systems adeptly handle tasks like fetching entire data files, answering factual queries, or performing targeted data processing, but typically do not focus on generating structured query language for in-depth interaction with relational geospatial databases.

2.3. Natural language interfaces for databases in the geospatial context

The development of NLIDB aims to simplify access to structured data for users unfamiliar with formal query languages like SQL. This pursuit faces inherent complexities in semantic parsing, schema matching to align natural language with database structures, and correct SQL syntax generation (Li and Jagadish 2014, Sutskever *et al.* 2014). Despite advances with pre-trained language model (Li *et al.* 2023), challenges such as syntax errors and schema mismatches persist (Kim *et al.* 2020), compounded by the "technical divide" in SQL proficiency (Zhao *et al.* 2024).

These NLIDB challenges are significantly amplified in the geospatial domain due to the complex and diverse nature of spatial databases, which include specialized structures (topological relations, coordinates, spatial indices) and require domain-specific operations (e.g., buffer analysis, spatial joins)(Li *et al.* 2024). A critical and pervasive hurdle is the semantic disambiguation of geographic names, which often derive their uniqueness only from specific contexts (Goodchild *et al.* 2005). High rates of ambiguity have been reported in various textual sources (Smith and Crane, 2001, Amitay *et al.* 2004, Garbin and Mani 2005), posing a major obstacle to accurate query interpretation.

Within this specialized sub-field, notable efforts such as NALSpatial (Liu *et al.* 2025) have advanced the field by converting natural language queries into custom 'executable languages' for spatial databases. However, the reliance on such proprietary intermediate languages can introduce limitations in terms of interoperability with standard database tools and workflows and may not fully harness the expressive power or broad compatibility of standard query languages. Consequently, the generation of standard SQL, coupled with robust handling of intricate geospatial schemas and effective, scalable resolution of geographic ambiguities, persists as a significant open challenge for developing truly accessible and reliable geospatial NLIDB.

2.4. Summary, gaps, and Monkuu's contributions

The review of existing literature reveals that while LLM has spurred considerable innovation in GIS, distinct gaps remain. Systems automating geospatial analysis or facilitating broad data retrieval and specialized tasks often do not address the nuanced challenges of precise, structured data extraction from complex geospatial databases via natural language. Even within the domain of NLIDB applied to geospatial contexts, robustly handling intricate spatial schemas, ensuring accurate generation of standard SQL, and effectively resolving pervasive geographic name ambiguities are persistent hurdles.

Monkuu is specifically designed to address these limitations by providing a more accurate and user-friendly natural language interface for geospatial databases. It achieves this through key innovations: (1) its generation of standard SQL enhances interoperability; (2) its dynamic context-aware schema mapping mechanism tackles complex schema alignment and mitigates information truncation issues in RAG methods, aiming for higher precision in query translation; and (3) its human-in-the-loop geographic disambiguation workflow offers a targeted solution for resolving ambiguous place names, a critical factor for usability. Collectively, these features enable Monkuu to deliver a more reliable data foundation for non-expert users and downstream geospatial applications, directly contributing to making complex geospatial data more accessible and actionable.

3. Methodology

The proposed methodology aims to translate natural language questions into executable SQL queries for geospatial databases, addressing challenges inherent in understanding spatial semantics and integrating domain-specific geographic information. This section details the system architecture, the core NL2SQL pipeline with a focus on geospatial prompt engineering, and the user-driven mechanism for disambiguating geographic entities. Human expertise is leveraged strategically: initially, domain experts verify and help construct standardized geospatial schema documentation, and during runtime, users participate in resolving geographic entity ambiguities, ensuring both accuracy and efficiency.

3.1. System overall architecture

Our system's architecture, illustrated in Figure 1, comprises a data processing workflow, middleware, an NL2SQL conversion method, and geospatial modules, distinguishing between implemented (solid) and extensible (dashed) components. A crucial preparatory step involves processing geospatial data with spatial indexing (e.g. R-tree) for optimized query performance, with indexed data stored in spatial databases. This foundation is critically supported by expert-verified, standardized schema documentation detailing essential GIS-specific information like feature geometry types and spatial attributes. This rich schema provides vital context for accurate interpretation by downstream components, including the Schema Mapper and SQL Generator. The core NL2SQL module executes key steps—intention recognition, geographic disambiguation,

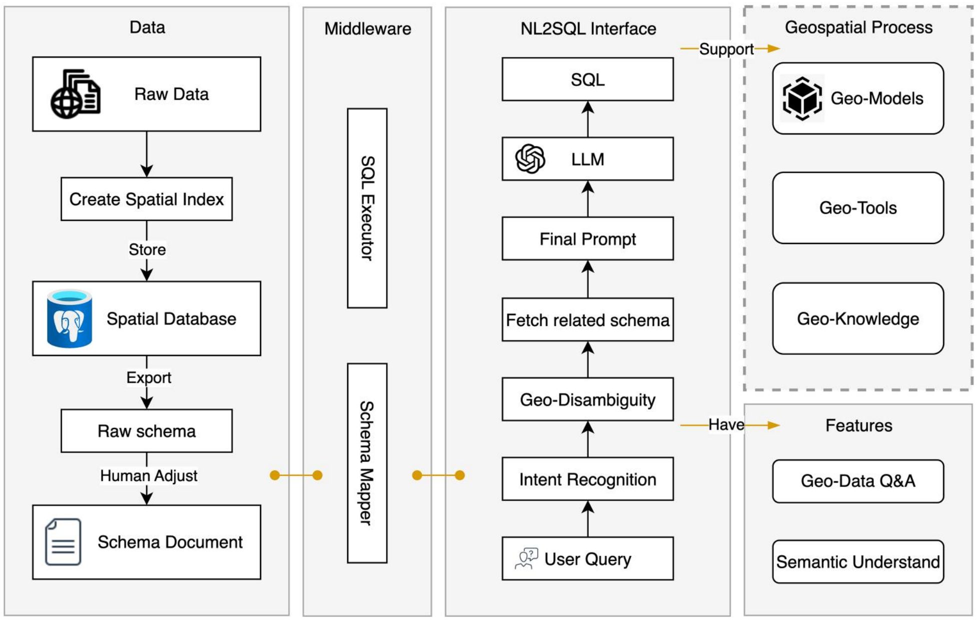


Figure 1. Architecture diagram of the proposed framework. The architecture includes the data processing workflow, middleware components, NL2SQL conversion method, geospatial processing modules, and system feature components. Solid components represent implemented contributions, dashed components indicate extensible interfaces.

schema acquisition, and SQL generation—and provides RESTful API interfaces, enabling its functionalities to be seamlessly embedded within other professional intelligent geospatial analysis software.

The Middleware Layer supports this by managing query execution through its SQL Executor (which includes permission controls) and collaborating with the Schema Mapper for precise natural language to database schema mapping. Human supervision is strategically integrated: domain experts create and validate the initial geospatial schema documentation, ensuring data integrity and semantic richness, while a human-in-the-loop approach is employed at runtime specifically for the nuanced task of geographic entity disambiguation, which remains challenging for full automation. Other pipeline operations, such as schema mapping and SQL generation, are automated, relying effectively on this verified schema and robust prompting techniques for efficiency.

The NL2SQL task, within this geospatial context, aims to generate an executable SQL query Y through a modeling process. This can be formally expressed as:

$$Y = f(\mathcal{Q}, \mathcal{S}, I | \theta) \tag{1}$$

In this equation, $\mathcal{Q}$ represents the natural language question posed by the user. The term $\mathcal{S}$ denotes the database schema information, which is structured as a triple $\mathcal{S} = \langle \mathcal{C}, \mathcal{T}, \mathcal{K} \rangle$. Here, $\mathcal{C}$ is the set of columns, $\mathcal{T}$ is the set of tables, and $\mathcal{K}$ signifies latent external knowledge. This external knowledge crucially includes the GIS-specific details derived from the standardized schema documentation. Finally, I represents system instructions, embodied as carefully designed prompts. These prompts guide the

LLM $f(\cdot|\theta)$ with parameters θ, to generate accurate SQL queries by providing explicit semantic guidance and task constraints, especially pertinent for geospatial operations.

3.2. Core NL2SQL pipeline with geospatial prompt engineering

Our spatial data retrieval method, the core of which is illustrated in Figure 2, comprises three critical modules: the Query Classifier, the GeoEntity Extractor, and the Schema Mapper. Following the operations of these modules, the SQL Generator produces the final query. A central principle of this pipeline is geospatial prompt engineering, a process wherein prompts are meticulously designed to imbue the LLM with an understanding of geographic concepts, spatial relationships, and specific database structures relevant to GIS.

3.2.1. User intent recognition

The Query Classifier(Figure2A), as the system's entry point, categorizes user queries to optimize workflow and route requests appropriately. Its primary goal is to classify natural language queries into predefined classes reflecting system capabilities, such as Geo_Query_Can_Solve for processable geographic queries, Chat_Query for conversational inputs, and Geo_Query_Cannot_Solve for queries outside its scope (e.g., data coverage), as illustrated in Table 1. The Large Language Model (LLM) receives a structured prompt containing a task instruction (e.g., "Analyze the following user query and classify its primary intent … ") along with category definitions and illustrative examples.

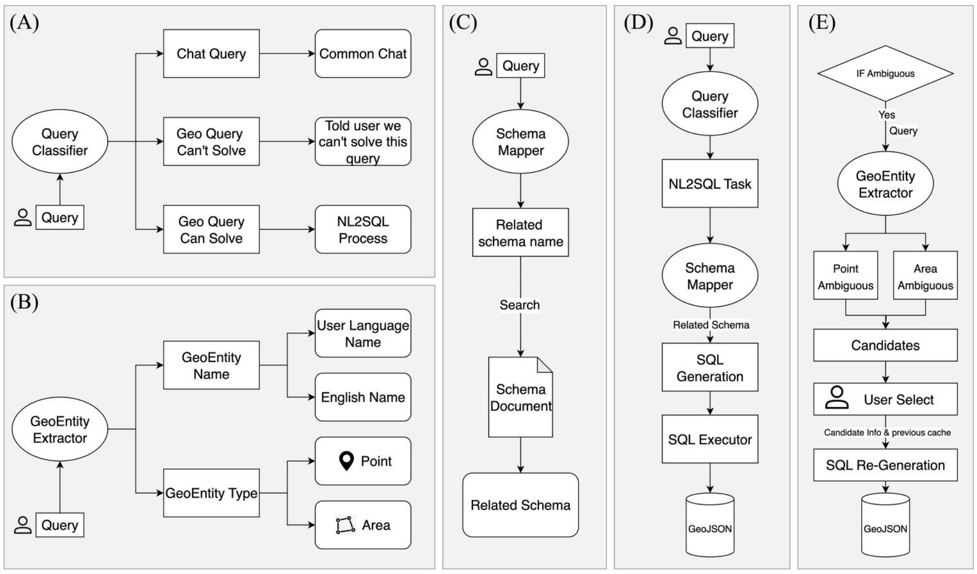


Figure 2. Schematic diagram of the core modules for spatial data retrieval. (A) Query Classifier module, designed to identify the semantic categories of queries; (B) GeoEntity Extractor module, responsible for extracting geographical entity information; (C) Schema Mapper module, which establishes semantic mapping relationships between queries and database schemas. (D) A typical NL2SQL workflow in Monkuu. (E) A typical Disambiguation workflow in Monkuu.

Table 1. Input and output examples of core NL2SQL modules (Query Classification, Geoname Extraction, and Schema Mapper).

ID	Query	Output_QC	Output_GE	Output_SM
1	Show me the Ōmiya Station	Geo_Query_Can_Solve	[{"entity": "Ōmiya Station", "name_ja": "大宮駅", "type": "Point"}]	["zenrin_poi", "railstation"]
2	How many data do you have?	Chat_Query	N/A	N/A
3	Can you show me POI in New York?	Geo_Query_Cannot_Solve	N/A	N/A

Table 1 demonstrates this classification: a query like "Show me the Ōmiya Station" is identified as Geo_Query_Can_Solve, proceeding to geospatial processing. In contrast, "How many data do you have?" is classified as a Chat_Query, and "Can you show me POI in New York?" results in Geo_Query_Cannot_Solve, this due to data limitations (e.g., "New York" being outside the current dataset). This classification enables effective routing: Geo_Query_Can_Solve queries engage the core NL2SQL pipeline, while other categories trigger appropriate handlers. This pre-processing concentrates resources, with prompt design principles (detailed further in supplementary Appendix) aiming for high reliability.

3.2.2. Geographic entity extraction

In Figure 2E, The GeoEntity Extractor is a key component of disambiguation workflow, it identifies and extracts geographic names and their likely types (e.g., Point, Area) from the user's query. The prompt for this module combines a GIS-focused Named Entity Recognition (NER) instruction with curated Geo_Examples that showcase typical geographic entities. The effectiveness of this module is also demonstrated in (Table 1, ID 1), where "Ōmiya Station" is extracted as a 'Point' entity with its Japanese name. Appendix 4 illustrates the prompt composition of the Geoname Extractor.

3.2.3. Schema mapping for geospatial databases

Providing sufficient contextual information for models is crucial in NL2SQL tasks (Fan *et al*. 2024). This requires precise descriptions of databases, including database names, table names, column names, and detailed textual descriptions. To address this challenge, this study proposes a novel workflow named Schema Mapper. The core of Schema Mapper is a single LLM call, whose prompt (detailed in Appendix 2) includes task-specific instructions, the schema abstract, and the user's query. Figure 2(C) illustrates the principle of the Schema Mapper, which serves as a key module in the framework to establish effective connections between user queries and existing database schemas. User queries are typically input in natural language. To accurately parse these queries, the Schema Mapper aims to identify and extract the table structure information from the system's databases that best matches the query's intent.

Considering the context window limitations of LLM (typically around 32K tokens), this study introduces a mechanism called "schema abstract" D_a. This mechanism provides a simplified yet sufficiently complete representation of database schemas, thereby maximizing the utilization of the LLM's available context window. Through this approach, it enables accurate extraction and matching of relevant table names within the given contextual constraints. Formula 2 is the mathematical representation of this module, where the natural language query Q, Database Abstract D_a and Insturction I are fed into LLM. Then process LLM response with Regular expression to obtain the relevant table names $T_{relevant}$

$$T_{relevant} = LLM(Q, D_a, I) \tag{2}$$

3.2.4. GIS-aware SQL generation

The final module in the pipeline is the SQL Generator, which produces the executable SQL query. This module employs a few-shot learning approach. Its comprehensive prompt $\mathcal{P}_0$ is constructed by concatenating several key components: an Instruction Component (I), the Related Schema ($\mathcal{S}_{related}$) identified by the Schema Mapper, a set of Few-Shot Examples, and the User Question ($\mathcal{Q}$) from the user.

$$\mathcal{P}_0 = I \oplus \mathcal{S}_{related} \oplus Q \tag{3}$$

As shown in the Prompt structure of Appendix 5, the instructions I are particularly important for GIS contexts, providing explicit guidance on the specific SQL dialect (e.g., PostGIS) and the appropriate use of spatial functions (e.g., "Use ST_DWithin for proximity queries, ST_Intersects for overlap checks"). The few-shot examples are carefully selected to showcase common geospatial query patterns and their corresponding correct SQL solutions, thereby providing the LLM with concrete templates for generating complex spatial queries. This structured, example-driven approach significantly enhances the model's ability to generate accurate and complex SQL queries suitable for geospatial databases.

3.3. Interactive disambiguation of geographic entities

Geographically named entity disambiguation primarily addresses the polysemy of place names across different contexts (Molina-Villegas *et al.* 2021). Resolving ambiguity in geographic named entities is a critical challenge for any NL2SQL system operating in the geospatial domain. Our system addresses this through a user-driven 'Human-in-the-Loop' mechanism. The typical workflow for this disambiguation process within Monkuu is illustrated in Figure 2(E). When there is a geographic entity E that is potentially ambiguous (e.g., multiple locations with the same name), the system initiates this interactive workflow. It first determines the likely type of the entity (point or area) and then invokes the appropriate disambiguation sub-module. A ranked list of candidate interpretations for E is subsequently presented to the user. The user's selection of the intended entity E^* (which then typically includes specific coordinates or a defined boundary) is fed back into the system to guide the generation of a precise and accurate query. For concrete examples, see Appendix 1.

$$\text{Disambiguate}(E) = \begin{cases} \text{GoogleMaps}(E) & \text{if } E \in \text{Point} \\ \text{RegionDB}(E) & \text{if } E \in \text{Area} \end{cases} \xrightarrow{\quad} \text{User Select} \quad E^* \tag{4}$$

3.3.1. Point-based disambiguation

For entities identified as point-like locations, such as specific addresses or POI, our system utilizes the Google Maps Autocomplete API. When a place name Q_p (extracted from the user query) is sent to this API G, it returns a list of candidate locations, each typically including a full name, address, and geographic coordinates. The user is then presented with these candidates and selects (σ) the one that matches their intent. This selection, E_p^*, provides the precise geographic coordinates needed for the query. While highly effective for disambiguating specific points, this method generally does not provide the detailed boundary information required for area-based entities.

$$E_p^* = \sigma(G(Q_p)) \tag{5}$$

3.3.2. Area-based disambiguation

To address ambiguities for area-based queries, which involve regions, administrative divisions, or other named geographical extents, we employ a custom-developed solution. This solution leverages a hierarchically structured toponym query tree and an associated regional database. This database is constructed using official boundary data, which is the chōme-level administrative unit data published by the Japanese government, ensuring high spatial accuracy.

When an area name Q_a is extracted from a query, our toponym query tree T is used to efficiently search the regional database. This process identifies potential candidate areas that match the user's input. These candidates are then presented to the user for selection (σ). Once the user confirms their target area, say D_j, an inverse mapping function M^{-1} is applied to retrieve the actual geometric boundary for D_j. This results in the precisely disambiguated area entity E_r^*. This custom approach, by utilizing precise administrative boundary data and an efficient querying mechanism, is crucial for accurately parsing complex regional queries where understanding the exact spatial extent is paramount for meaningful GIS analysis.

$$\begin{cases} D_j = \sigma(T(Q_a)) \\ E_r^* = M^{-1}(D_j) \end{cases} \tag{6}$$

3.4. Evaluation experiment design

3.4.1. Evaluation dataset

(1) KaggleDBQA. KaggleDBQA (Lee *et al.* 2021) is a cross-domain NL2SQL benchmark dataset specifically designed for semantic parsing tasks in real-world scenarios. Built upon unnormalized real web databases, this dataset encompasses complex queries and natural language questions to simulate practical question-answering contexts. Compared to existing datasets, KaggleDBQA preserves databases' original formats, generates questions through naturalistic environments, and provides rich domain-specific documentation to support semantic parsing. Table 2 demonstrates the dataset's detailed composition, KaggleDBQA contains 8 databases with 272 test instances, averaging 2.25 subtables per database, exhibiting significant complexity and real-world applicability. This study employs this dataset to evaluate Monkuu's conventional NL2SQL performance.

(2) GeoQueryJP. Current academic research lacks standardized datasets designed explicitly for evaluating geographic disambiguation performance. This study constructs GeoQueryJP (detailed in Table S1), a geographic data query accuracy test set. The

Table 2. Description of the KaggleDBQA dataset.

Metric	Value
#Examples	272
#DB	8
#Table/DB	2.25
%WHERE	8.7
%VAL	73.5
%SELECT	24.6
%NON-SELECT	6.8

%WHERE measures the percentage of examples where all WHERE/HAVING columns in SQL queries are explicitly mentioned in corresponding natural language questions. %VAL evaluates the coverage of all values in SQL queries; %SELECT compares all SELECT columns; %NON-SELECT assesses all columns, excluding those in SELECT clauses.

Table 3. Overview of the geographic disambiguation test dataset.

Type	Number	Example
Multiple POIs with the same name	2	Ōmiya Station
Variation in notation	4	霞が関 and 霞ヶ関
Cities市 with the same name	4	Fuchū city
Wards区 with the same name	5	Chūō City
Districts郡 with the same name and same pronunciation	2	Aki District
Districts郡 with the same name but different pronunciation	3	Aichi District in Aichi
Towns町 with the same name	5	Oguni town
Regions with the same name but different administrative divisions	14	Fukushima
Same regions with different name	14	アキバ and 秋葉原

Type indicates the classification category of queries, *Number* specifies the number of such queries in the dataset, and *Example* demonstrates representative geographic entities contained in this query type.

dataset is developed based on Wikipedia's 'List of ambiguous geographic names', with all entries confined to Japanese administrative divisions due to data collection constraints. As shown in Table 3, the GeoQueryJP dataset comprises 53 test instances covering 9 prototypical geographic ambiguity types. Each category is systematically characterized through three dimensions: type classification, instance quantity, and exemplary entities.

3.4.2. Evaluation metrics

(1) Execution accuracy. In real-world application scenarios, the ability of NL2SQL to accurately and effectively deliver expected results is prioritized. Therefore, this study employs the most commonly used metric in traditional NL2SQL research, Execution Accuracy (EX), to evaluate the system. EX measures the correctness of predicted SQL queries by executing them in the corresponding database and comparing the execution results with those obtained from GroundTruth queries. EX can be calculated using Formula 7, where V_n represents the Ground Truth for the *n-th* query, $\widehat{V_n}$ denotes the execution result of the SQL generated by this system, and $\mathbb{1}\left(V_n, \widehat{V_n}\right)$ is an indicator function that assigns a value of 1 when the results match the Ground Truth and 0 otherwise.

$$\begin{cases} \text{EX} = \dfrac{\sum_{n=1}^{N} \mathbb{1}\left(V_n, \widehat{V_n}\right)}{N} \\ \mathbb{1}\left(V, \widehat{V_n}\right) = \begin{cases} 1, & V = \hat{V} \\ 0, & V \neq \hat{V} \end{cases} \end{cases} \tag{7}$$

(2) Geographic ambiguity Resolution recall. Monkuu addresses geographic ambiguity issues by presenting users with candidate options for selection and introducing human-system interaction. Drawing on the calculation methodology of EX, this study proposes Geographic Ambiguity Resolution Recall (GARR) to quantify the system's capability in handling geographic ambiguity, as shown in Formula 8. A disambiguation process is deemed standardized and correct if the candidate set contains the true location referenced in the user's query. Let a denote the ground-truth location and A represent the candidate location set.

$$\begin{cases} \mathrm{GARR} = \dfrac{\sum_{n=1}^{N} \mathbb{1}(a_n, A_n)}{N} \\ \mathbb{1}(a, A) = \begin{cases} 1, & a \in A \\ 0, & a \notin A \end{cases} \end{cases} \tag{8}$$

4. Results

4.1. Module accuracy and comparison

The experimental evaluation encompasses performance testing in two key aspects: the accuracy of geographic entity disambiguation and the effectiveness of translating natural language to SQL for traditional NL2SQL tasks.

Table 4 presents results for geographic query testing on the GeoQueryJP dataset. Without its disambiguation module (w/o Disambiguation Module), the system achieved a Geographic Ambiguity Resolution Recall (GARR) of 40.3%. Upon integrating Monkuu's human-in-the-loop disambiguation module, the GARR increased to 82.4%, a rise of 42.1 percentage points.

To further dissect the contribution of Monkuu's core components and compare different strategies, an ablation study was conducted. Table 5 summarizes the results of this study, evaluating the impact of the Schema Mapper and the Disambiguation Module on their respective tasks.

The ablation study results in Table 5 clearly highlight the individual contributions of Monkuu's key architectural components. For NL2SQL tasks on KaggleDBQA, the full Monkuu system (utilizing GPT-4o-mini with its Schema Mapper) achieved an EX of 56.2%. When the Schema Mapper was replaced with a RAG approach for schema provision, the accuracy dropped to 50.3%. More strikingly, when the Schema Mapper was removed and the complete database schema was directly provided to the LLM (Full Schema), the EX accuracy plummeted to 16.2%.

Moving to a broader comparison with existing NL2SQL models, Table 6 displays the SQL execution accuracy (EX) on the KaggleDBQA dataset. Traditional neural network

Table 4. Monkuu's accuracy performance in geographic testing, is divided into two phases: *Initial Test* indicates the accuracy of the initial query; *Disambiguation Added* represents the matching accuracy between candidate geographic entities generated by activating the disambiguation workflow (after initial query errors) and the true geographic entities in the query.

Test Phase	Correct/Total	GARR(%)
Initial Test	23/57	40.3%
Disambiguation Added	47/57	82.4%

Table 5. Ablation study of Monkuu components.

System Configuration	Dataset	Metric	Score (%)
Monkuu (Full System)	KaggleDBQA	EX	56.2
w/o Schema Mapper (using Full Schema)	KaggleDBQA	EX	16.2
w/o Schema Mapper (using RAG)	KaggleDBQA	EX	50.3
Monkuu (Full System)	GeoQueryJP	GARR	82.4
w/o Disambiguation Module	GeoQueryJP	GARR	40.3

Table 6. Evaluation performance (%) of different NL2SQL models on KaggleDBQA.

Models	EX
EditSQL (Zhang *et al.* 2019)	11.7
RAT-SQL (Wang *et al.* 2019)	13.6
RESDSQL (Li *et al.* 2023)	31.9
ZeroNL2SQL (Fan *et al.* 2024)	42.4
Monkuu	**56.2**

This study compares Monkuu with baseline models and existing SOTA models in the evaluation phase.

Table 7. Error type counts of Monkuu using different LLM under varying schema information acquisition methods on KaggleDBQA.

Error Type/Model(Modules)	DeepseekV3 (Schema Mapper)	GPT4o-mini (RAG)	GPT4o-mini (Schema Mapper)	GPT4o-mini (Full-Schema)
SQL Execution Error	42	25	34	152
Correct	144	137	153	44
Row Count Mismatch	44	44	39	12
Data Content Mismatch	42	65	45	64
EX(%)	**52.9**	**50.3**	**56.2**	**16.2**

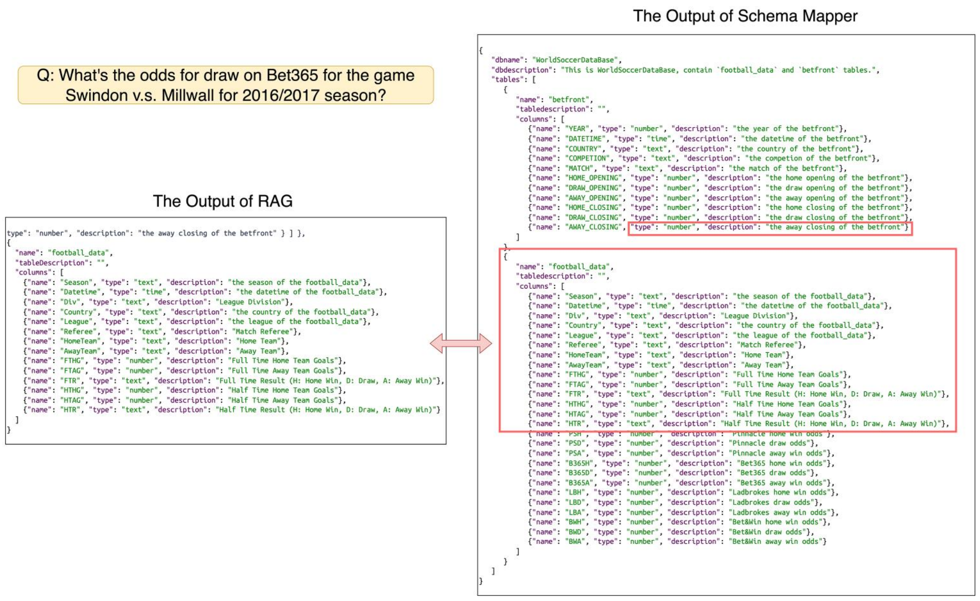
models like EditSQL (Zhang *et al.* 2019) and RAT-SQL (Wang *et al.* 2019) achieved accuracies of 11.7% and 13.6%, respectively. The pre-trained language model (PLM)-based RESDSQL (Li *et al.* 2023) attained 31.9% accuracy. Among LLM-based approaches, ZeroNL2SQL (Fan *et al.* 2024) reached 42.4%. In contrast, our Monkuu system, with its sophisticated schema handling and LLM integration, achieved the highest accuracy of 56.2%.

Table 7 details error types and accuracy for Monkuu under different configurations on the KaggleDBQA dataset. Monkuu (GPT-4o-mini with Schema Mapper) achieved 56.2% EX, with 34 SQL execution errors and 45 data content mismatches. The RAG approach (GPT-4o-mini) yielded 50.3% EX, with 25 SQL execution errors and 65 data content mismatches. The DeepseekV3 model (with Schema Mapper) achieved 52.9% EX. The Full-Schema baseline (GPT-4o-mini with entire schema) resulted in 16.2% EX, with 152 SQL execution errors and 64 data content mismatches.

Figure 3 illustrates a comparison of output results from the Schema Mapper and RAG for a sample query, showing the retrieved schema components.

4.2. Case study

To illustrate the practical application and efficacy of Monkuu in translating natural language queries into executable SQL for geospatial databases, this subsection presents two distinct case studies. These examples, depicted in Figure 4, showcase Monkuu's


Q: What's the odds for draw on Bet365 for the game Swindon v.s. Millwall for 2016/2017 season?
The Output of RAG
The Output of Schema Mapper

```
type": "number", "description": "the away closing of the betfront" } ] },
{
  "name": "football_data",
  "tableDescription": "",
  "columns": [
    {"name": "Season", "type": "text", "description": "the season of the football_data"},
    {"name": "Datetime", "type": "time", "description": "the datetime of the football_data"},
    {"name": "Div", "type": "text", "description": "League Division"},
    {"name": "Country", "type": "text", "description": "the country of the football_data"},
    {"name": "League", "type": "text", "description": "the league of the football_data"},
    {"name": "Referee", "type": "text", "description": "Match Referee"},
    {"name": "HomeTeam", "type": "text", "description": "Home Team"},
    {"name": "AwayTeam", "type": "text", "description": "Away Team"},
    {"name": "FTHG", "type": "number", "description": "Full Time Home Team Goals"},
    {"name": "FTAG", "type": "number", "description": "Full Time Away Team Goals"},
    {"name": "FTR", "type": "text", "description": "Full Time Result (H: Home Win, D: Draw, A: Away Win)"},
    {"name": "HTHG", "type": "number", "description": "Half Time Home Team Goals"},
    {"name": "HTAG", "type": "number", "description": "Half Time Away Team Goals"},
    {"name": "HTR", "type": "text", "description": "Half Time Result (H: Home Win, D: Draw, A: Away Win)"}
  ]
}
```

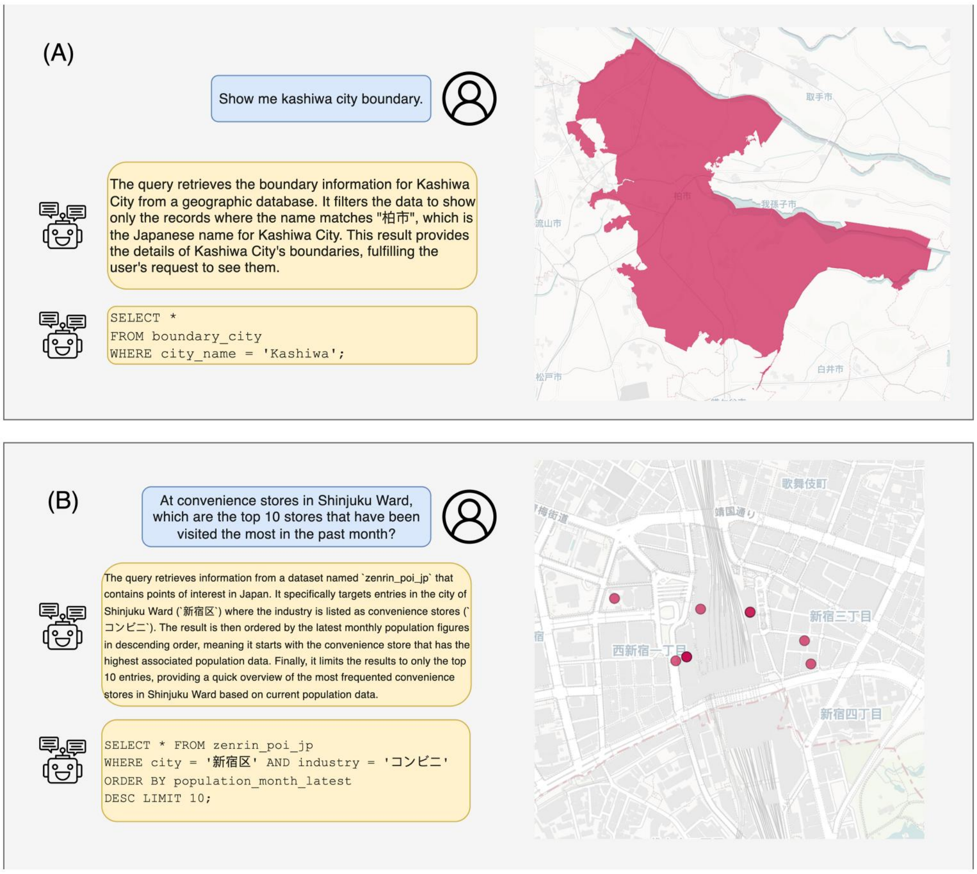
```
{
   "dbname": "WorldSoccerDataBase",
   "dbdescription": "This is WorldSoccerDataBase, contain `football_data` and `betfront` tables.",
   "tables": [
      {
         "name": "betfront",
         "tabledescription": "",
         "columns": [
            {"name": "YEAR", "type": "number", "description": "the year of the betfront"},
            {"name": "DATETIME", "type": "time", "description": "the datetime of the betfront"},
            {"name": "COUNTRY", "type": "text", "description": "the country of the betfront"},
            {"name": "COMPETION", "type": "text", "description": "the competion of the betfront"},
            {"name": "MATCH", "type": "text", "description": "the match of the betfront"},
            {"name": "HOME_OPENING", "type": "number", "description": "the home opening of the betfront"},
            {"name": "DRAW_OPENING", "type": "number", "description": "the draw opening of the betfront"},
            {"name": "AWAY_OPENING", "type": "number", "description": "the away opening of the betfront"},
            {"name": "HOME_CLOSING", "type": "number", "description": "the home closing of the betfront"},
            {"name": "DRAW_CLOSING", "type": "number", "description": "the draw closing of the betfront"},
            {"name": "AWAY_CLOSING", "type": "number", "description": "the away closing of the betfront"}
         ]
      },
      {
         "name": "football_data",
         "tabledescription": "",
         "columns": [
            {"name": "Season", "type": "text", "description": "the season of the football_data"},
            {"name": "Datetime", "type": "time", "description": "the datetime of the football_data"},
            {"name": "Div", "type": "text", "description": "League Division"},
            {"name": "Country", "type": "text", "description": "the country of the football_data"},
            {"name": "League", "type": "text", "description": "the league of the football_data"},
            {"name": "Referee", "type": "text", "description": "Match Referee"},
            {"name": "HomeTeam", "type": "text", "description": "Home Team"},
            {"name": "AwayTeam", "type": "text", "description": "Away Team"},
            {"name": "FTHG", "type": "number", "description": "Full Time Home Team Goals"},
            {"name": "FTAG", "type": "number", "description": "Full Time Away Team Goals"},
            {"name": "FTR", "type": "text", "description": "Full Time Result (H: Home Win, D: Draw, A: Away Win)"},
            {"name": "HTHG", "type": "number", "description": "Half Time Home Team Goals"},
            {"name": "HTAG", "type": "number", "description": "Half Time Away Team Goals"},
            {"name": "HTR", "type": "text", "description": "Half Time Result (H: Home Win, D: Draw, A: Away Win)"},
            {"name": "PSH", "type": "number", "description": "Pinnacle home win odds"},
            {"name": "PSD", "type": "number", "description": "Pinnacle draw odds"},
            {"name": "PSA", "type": "number", "description": "Pinnacle away win odds"},
            {"name": "B365H", "type": "number", "description": "Bet365 home win odds"},
            {"name": "B365D", "type": "number", "description": "Bet365 draw odds"},
            {"name": "B365A", "type": "number", "description": "Bet365 away win odds"},
            {"name": "LBH", "type": "number", "description": "Ladbrokes home win odds"},
            {"name": "LBD", "type": "number", "description": "Ladbrokes draw odds"},
            {"name": "LBA", "type": "number", "description": "Ladbrokes away win odds"},
            {"name": "BWH", "type": "number", "description": "Bet&Win home win odds"},
            {"name": "BWD", "type": "number", "description": "Bet&Win draw odds"},
            {"name": "BWA", "type": "number", "description": "Bet&Win away win odds"}
         ]
      }
   ]
}
```

Figure 3. Comparison of output results between Schema Mapper and RAG for the query: 'What's the odds for draw on Bet365 for the game Swindon vs. Millwall for the 2016/2017 season?' in a testing environment. The output from RAG, highlighted by the red rectangle, is encompassed within the more comprehensive output from Schema Mapper.

workflow from understanding user intent to retrieving and visualizing pertinent geospatial information, thereby highlighting its core functionalities. The cases presented in this section were executed once through the Monkuu system, accurately obtaining the expected results.

4.2.1. Geospatial boundary retrieval

The first case, presented in Figure 4(A), begins with a user issuing a direct natural language query: 'Show me Kashiwa City boundary.' This query requests the retrieval of a specific administrative geospatial feature. Monkuu processes this input by first identifying the core intent—to display the boundary of 'Kashiwa City.' As detailed in the system's explanation within Figure 4(A), Monkuu accesses a geographic database and filters records to isolate the entry corresponding to 'Kashiwa City' (柏市). This interpretation leads to the generation of an SQL query, also shown in Figure 4(A), which selects all attributes from the boundary_city table where the city_name field matches 'Kashiwa'. The successful execution of this query results in the map visualization of Kashiwa City's administrative boundary, directly fulfilling the user's request. This example, while straightforward, demonstrates Monkuu's fundamental capability to parse simple NL queries, map recognized entities to the correct database schema—implicitly utilizing the Schema Mapper to associate "city boundary" with the boundary_city table and "Kashiwa City" with the city_name attribute—and render spatial data. While not explicitly invoked by this simple query, Monkuu's geographic disambiguation module would be engaged if "Kashiwa City" presented ambiguity within a broader dataset.


(A)

Show me kashiwa city boundary.

The query retrieves the boundary information for Kashiwa City from a geographic database. It filters the data to show only the records where the name matches "柏市", which is the Japanese name for Kashiwa City. This result provides the details of Kashiwa City's boundaries, fulfilling the user's request to see them.

```
SELECT *
FROM boundary_city
WHERE city_name = 'Kashiwa';
```

(B)

At convenience stores in Shinjuku Ward, which are the top 10 stores that have been visited the most in the past month?

The query retrieves information from a dataset named `zenrin_poi_jp` that contains points of interest in Japan. It specifically targets entries in the city of Shinjuku Ward (`新宿区`) where the industry is listed as convenience stores (`コンビニ`). The result is then ordered by the latest monthly population figures in descending order, meaning it starts with the convenience store that has the highest associated population data. Finally, it limits the results to only the top 10 entries, providing a quick overview of the most frequented convenience stores in Shinjuku Ward based on current population data.

```
SELECT * FROM zenrin_poi_jp
WHERE city = '新宿区' AND industry = 'コンビニ'
ORDER BY population_month_latest
DESC LIMIT 10;
```

Figure 4. Monkuu processing natural language geospatial queries: (A) Kashiwa City boundary retrieval, showing user query, system interpretation, generated SQL, and map result. (B) Query for top 10 Shinjuku convenience stores by recent popularity, detailing user query, system interpretation involving semantic mapping, generated SQL, and visualized POI results.

4.2.2. Querying POI with multiple constraints and semantic mapping

The second case study, illustrated in Figure 4(B), involves a more intricate natural language query: 'At convenience stores in Shinjuku Ward, which are the top 10 stores that have been visited the most in the past month?' This query requires identifying specific Points of Interest (POIs), 'convenience stores,' within a defined geographic area, 'Shinjuku Ward,' applying a ranking based on a nuanced temporal concept, 'visited the most in the past month,' and limiting the output to the "top 10" results.

Monkuu's approach to this query, as explained by the system in Figure 2(B), involves several steps. Initially, it targets the zenrin_poi_jp dataset, filtering entries located in 'Shinjuku Ward' (新宿区) and categorized under the 'convenience stores' (コンビニ) industry. A critical aspect of this case is Monkuu's semantic interpretation capability, facilitated by the Schema Mapper. The phrase 'visited the most in the past month' is semantically mapped to the database column population_month_latest, which is then used to order the results in descending order. This demonstrates Monkuu's ability to bridge the semantic gap between potentially abstract or indirect user language and concrete database field names. Subsequently, the system limits the retrieved POIs to the top 10, fulfilling all aspects of the user's request. The SQL query generated by Monkuu to achieve this, incorporating WHERE clauses for location and industry, an ORDER BY clause for the inferred popularity metric, and a LIMIT clause, is presented in Figure 4(B). The resulting map visualization displays the locations of these top 10 convenience stores. This case effectively showcases Monkuu's proficiency in deconstructing multi-faceted natural language queries, performing accurate entity and attribute filtering (including the identification of "Shinjuku Ward," which would involve disambiguation if necessary), and utilizing its Schema Mapper for intelligent semantic mapping to generate precise SQL.

Collectively, these case studies underscore Monkuu's ability to serve as an effective natural language interface for geospatial databases, adeptly handling both direct data retrieval requests and more complex queries requiring deeper semantic understanding and structured data manipulation.

5. Discussion

The inherent complexity of structured query languages and geospatial data schemas has long presented a significant barrier to accessing valuable information stored in Geographic Information Systems, especially for non-expert users. This study introduced Monkuu, a natural language interface specifically engineered to address these challenges for geospatial databases. As an interface, Monkuu's primary contribution lies in its robust database access capabilities, providing clean data interfaces for downstream spatial analysis tools while maintaining focus on accurate query translation. By integrating a dynamic context-aware Schema Mapper and a human-in-the-loop geographic disambiguation workflow, Monkuu aims to make geospatial data more discoverable and usable through intuitive natural language interaction.

5.1. Advancements in natural language access to geospatial databases

Monkuu's core technical contributions substantially enhance the accuracy and practicality of natural language interfaces for geospatial databases, validating its innovative design choices. A central achievement is its strong performance in NL2SQL tasks, marked by a 56.2% execution accuracy on the KaggleDBQA benchmark. This result not only surpasses the leading ZeroNL2SQL model by a significant margin of 13.8 percentage points but also demonstrates a considerable advancement over earlier NLIDB approaches, underscoring Monkuu's superior capability in translating complex natural language questions into executable SQL in real-world scenarios.

This enhanced accuracy in query translation is critically supported by the innovative Schema Mapper. The importance of the Schema Mapper was clearly evidenced in our ablation studies. Its ability to dynamically select and provide concise, relevant schema information to the LLM led to superior performance (56.2% EX) when compared to both a standard RAG-based schema retrieval method (50.3% EX) and, most notably, a Full-Schema baseline (16.2% EX). The RAG approach, while an improvement over no schema information, can often struggle with information truncation or incomplete retrieval from extensive database documentation, potentially leading to errors in understanding complex schemas. The drastically lower performance of the Full-Schema baseline is particularly telling: it robustly demonstrates that for effective NL2SQL, an intelligent mechanism to manage schema complexity is paramount. Providing an LLM with excessive, unfiltered schema information leads to severe information overload, significantly impairing its ability to generate correct SQL queries. The Schema Mapper, therefore, is not merely a tool for token efficiency but a vital component. It ensures the LLM can accurately interpret the database structure, fundamentally addressing the issues of both schema information truncation (common in RAG, Figure 3) and information overload (in Full-Schema approaches), thereby facilitating precise query translation. This finding strongly validates the necessity of such an intelligent schema filtering and mapping mechanism.

For an interface to geospatial databases to be truly effective, it must also capably handle the pervasive issue of geographic name ambiguity. Monkuu's human-in-the-loop geographic disambiguation workflow addresses this directly, improving the GARR on the GeoQueryJP dataset from a baseline of 40.3% to 82.4%. This ensures that the data retrieved through the natural language interface accurately reflects the user's intended geographic focus, which is essential for providing a "clean data interface." By generating standard SQL, Monkuu further ensures that the data retrieved can be seamlessly used by a wide array of standard database tools and downstream applications, enhancing interoperability. The case studies presented (Section 4.2), such as processing queries like "Show me the boundary of Kashiwa City" or identifying specific Points of Interest in Shinjuku Ward into executable SQL, effectively demonstrate these capabilities in action.

5.2. Implications for geospatial data accessibility and use

The advancements embodied in Monkuu have significant implications for how users can interact with complex geospatial data. By substantially lowering the technical barriers associated with SQL and intricate database schemas, Monkuu empowers a broader range of users, including domain experts who may not be GIS or database specialists, to directly query geospatial information using natural language. This democratization of access is crucial for leveraging the full potential of geospatial data in diverse fields.

Monkuu's focus on providing "clean data interfaces for downstream spatial analysis tools" is a key aspect of its utility. While Monkuu itself does not perform spatial analysis, its ability to accurately translate natural language requests into SQL queries ensures that subsequent analytical processes, whether conducted by users manually or by other specialized software, are based on correctly retrieved and relevant data. This accurate data provisioning step is fundamental to the integrity of any ensuing analysis or decision-making. By simplifying this initial, yet critical, data access stage, Monkuu contributes to advancing equitable and accessible geographic information services, allowing users to focus more on interpreting information and less on the mechanics of data retrieval.

5.3. Limitations

Despite its contributions, Monkuu has several limitations that should be acknowledged, impacting its performance, autonomy, and overall applicability. A primary factor influencing Monkuu's performance is its dependency on the quality and detail of available schema documentation. The Schema Mapper's accuracy in translating natural language to SQL is inherently linked to this documentation; thus, in scenarios where it is sparse, poorly maintained, or highly unconventional, the system's ability to correctly map user queries to the appropriate database schema may be significantly diminished. This reliance underscores a challenge for deploying Monkuu in environments with less mature data governance.

Limitations also arise within the geospatial entity disambiguation process. This process is not yet fully automated; while the human-in-the-loop component enhances accuracy for ambiguous cases, it can affect throughput in applications requiring rapid, unattended query processing. Furthermore, point-based disambiguation relies on external APIs like Google Maps, introducing dependencies related to service availability, potential costs, and data freshness.

Beyond disambiguation challenges, the precision of Monkuu's SQL generation for complex queries is also closely tied to the comprehensiveness and accuracy of the schema documentation, a dependency noted earlier. This is illustrated by its handling of the query, "How many nursery schools are there in Adachi city?", for which the system generated SELECT * FROM zenrin_poi_jp WHERE city='Adachi' AND industry='教育機関'; While the system correctly identified the broader '教育機関' (educational institution) category, its failure to apply a specific filter for "nursery schools" (e.g., name_full LIKE '%幼稚園%) can be attributed to a lack of explicit guidance within the schema documentation on how to distinguish such fine-grained entity sub-types or map them to specific filtering conditions. Similarly, potential mismatches in geographic name representation (e.g., the system using city='Adachi' when the database might expect the Japanese form '足立区') can also arise if the schema documentation does not sufficiently detail expected data formats or normalization rules. In this instance, the query also revealed a separate challenge in interpreting the aggregation intent, as the system performed a record retrieval (SELECT *) instead of the requested count (COUNT(*)). This highlights that while enriched schema documentation can significantly improve the mapping of specific entities and values, fully parsing all elements of complex query intent, such as implicit aggregations, remains an ongoing challenge for LLM-based systems.

Finally, Monkuu's current scope and generalizability are constrained in terms of geographic and linguistic coverage. The custom regional database integral to its area-based disambiguation is geographically limited, primarily covering Japan, which restricts its direct applicability for resolving area-based ambiguities in other regions. Moreover, the system's effectiveness has been predominantly demonstrated using English language queries; its performance with other languages has not yet been systematically evaluated, limiting its broader international utility at present.

5.4. Future work

To broaden Monkuu's geographic applicability and address limitations tied to region-specific resources, a crucial step will be to revamp the area-based disambiguation module. Our proposed approach involves integrating a global administrative boundaries database, such as GADM, with a geocoding service like the Google Geocoding API. When the Geoname Extractor identifies a potential area name, the Geocoding API will be used to obtain its likely geographic coordinates. These coordinates will then be spatially intersected with the GADM dataset to identify the specific administrative polygon the user is referring to. This workflow will replace the current Japan-specific regional database, enabling more robust and global area disambiguation. This initiative will also involve evaluating Monkuu's performance on multi-regional benchmarks to ensure its core capabilities generalize effectively across diverse geographical contexts.

Further research will empower users to work with their own data by investigating mechanisms for handling user-uploaded geospatial datasets. This could involve developing an interactive process where Monkuu assists users in defining, understanding, or mapping the schema of new, unfamiliar data sources, thereby expanding its utility beyond pre-defined databases and fostering greater flexibility.

To improve integration with downstream spatial analysis tools and other diverse systems, we will explore the adoption of standardized communication protocols. Specifically, future work will investigate the design and implementation of an interface compliant with Model Context Protocol (MCP), potentially culminating in Monkuu offering an MCP server. This would allow various client applications to seamlessly consume structured data and contextual information from Monkuu, positioning it as a robust hub in a broader geospatial data ecosystem. Future plans also include dedicated efforts to improve multilingual generalization capabilities, enabling Monkuu to support a wider global audience.

6. Conclusion

This study proposes Monkuu, an LLM-based natural language interface designed to enhance accessibility to geospatial databases. Monkuu achieves state-of-the-art performance on NL2SQL tasks, attaining an accuracy of 56.2% on the KaggleDBQA dataset, significantly outperforming existing approaches and underscoring its capability for accurate query translation. By effectively enabling LLM to connect with structured external geographic data sources, Monkuu addresses a critical information access challenge. Central to its success are two key innovations: a user-driven interactive disambiguation workflow, which substantially enhances the system's ability to resolve ambiguity in geographic place names, thereby ensuring the correct data is targeted; and a dynamic, context-driven schema mapping strategy. This Schema Mapper mitigates the low recall issues inherent in traditional RAG methods and addresses knowledge alignment bottlenecks between LLM and spatial databases, which is fundamental for precise NL2SQL conversion. The practical effectiveness of Monkuu in providing clean data interfaces is validated through illustrative case studies, which demonstrate its ability to process complex natural language queries into executable

SQL suitable for downstream applications, such as retrieving data for understanding site characteristics or informing location-based analyses.

Future research will focus on extending Monkuu's capabilities by: improving its global geographic disambiguation through the integration of global boundary datasets and geocoding services; enabling support for user-uploaded datasets via interactive schema mapping to enhance flexibility; and boosting system integration by exploring standardized communication protocols, such as the Model Context Protocol (MCP), for seamless interoperability. These advancements are designed to establish Monkuu as a more robust, adaptable, and broadly connected natural language interface for accessing geospatial information. These improvements aim to solidify Monkuu's effectiveness and flexibility, making geospatial data more accessible to all.

Disclosure statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and codes availability statement

Both the Monkuu code and its evaluation dataset are publicly available at: https://doi.org/10.6084/m9.figshare.28573940.v1

Funding

This work was supported by the National Natural Science Foundation of China [42471491, 42171466], the National Key Research and Development Program of China [2023YFB3906803], the Key Laboratory of Earth Surface System and Human-Earth Relations, Ministry of Natural Resources of China [LBXT2023YB03], and the "CUG Scholar" Scientific Research Funds at China University of Geosciences (Wuhan) [2022034]. This work was partly supported by the Project Grant from the Co-creation Center for Disaster Resilience, IRIDeS, Tohoku University (ID: 2-QR001).

Notes on contributors

Chenglong Yu is a graduate student at China University of Geosciences (Wuhan), China and an intern student at LocationMind Institute, LocationMind Inc., Japan. His research interests include GeoAI and Large Language Model.

Yao Yao is a Professor at China University of Geosciences (Wuhan), Hitotsubashi University, and Reitaku University. He has previously served as a Project Researcher at the University of Tokyo. His research interests include spatiotemporal big data mining, social geographic computing, and urban geographic information systems.

Xiang Zhang is a graduate student at China University of Geosciences (Wuhan), China and an intern student at LocationMind Institute, LocationMind Inc., Japan. His research interests include GeoAI and human mobility.

Geyuan Zhu is a graduate student at China University of Geosciences (Wuhan), China and an intern student at LocationMind Institute, LocationMind Inc., Japan. His research interests are intelligent agriculture, and large language model.

Yanduo Guo is an undergraduate student at China University of Geosciences(Wuhan), China and an intern student at LocationMind Institute, LocationMind Inc., Japan. His research interests include GeoAI and retrieval-augmented large language models.

Xiaowei Shao is an Associate Professor in the Faculty of Engineering at Reitaku University, Japan. His main research interests machine intelligence, pattern recognition, and AI-reasoning.

Mariko Shibasaki is a Consultant at LocationMind Institute, LocationMind Inc., Japan. She received the master degree from the Graduate School of Frontier Sciences, the University of Tokyo. Her interest is application geospatial foundation models to sustainable and inclusive development involved with human society and the natural environment.

Zhihui Hu is a graduate student at China University of Geosciences (Wuhan), China. His research interests are geospatial big data mining and geospatial foundation modeling.

Liangyang Dai is a graduate student at China University of Geosciences (Wuhan). His research interests are geospatial big data mining and health geography.

Qingfeng Guan is a Professor at China University of Geosciences (Wuhan). His research interests include high-performance spatial intelligence computation and urban computing.

Ryosuke Shibasaki is a Project Professor at the School of Interdisciplinary Information Studies at the University of Tokyo, Japan. His research interests cover mobile big data analysis, satellite/ aerial imagery and sensor data analysis, including automated mapping with deep learning, human behavior modeling/simulation, and data assimilation of discrete moving objects.

References

Achiam, J., *et al.*, 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774.*

Akinboyewa, T., *et al.*, 2024. GIS Copilot: Towards an Autonomous GIS Agent for Spatial Analysis. *arXiv preprint arXiv:2411.03205.*

Amitay, E., *et al.*, 2004. Web-a-where: geotagging web content. *In*: *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval,* Sheffield, United Kingdom. New York, NY: Association for Computing Machinery, 273–280.

Boiko, D.A., *et al.*, 2023. Autonomous chemical research with large language models. *Nature*, 624 (7992), 570–578.

Church, K.W., 2017. Word2Vec. *Natural Language Engineering*, 23 (1), 155–162.

Devlin, J., *et al.*, 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In:* Burstein, J., *et al.*, eds, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics, 4171–4186.

Fan, J., *et al.*, 2024. Combining small language models and large language models for zero-shot NL2SQL. *Proceedings of the VLDB Endowment*, 17 (11), 2750–2763.

Garbin, E., and Mani, I., (2005). Disambiguating toponyms in news. *In*: *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, Vancouver, British Columbia, Canada. USA: Association for Computational Linguistics, 363–370.

Goodchild, M.F., *et al.*, 2005. *Geographic information systems and science. Wiley & Sons, West Sussex, UK*, **17**, 517.

Gupta, D.V., *et al.*, 2024. Geode: a zero-shot geospatial question-answering agent with explicit reasoning and precise spatio-temporal retrieval. *arXiv preprint arXiv:2407.11014.*

Hu, Y., *et al.*, 2023. Geo-knowledge-guided GPT models improve the extraction of location descriptions from disaster-related social media messages. *International Journal of Geographical Information Science*, 37 (11), 2289–2318.

Huang, C., *et al.*, 2024. GeoAgent: to empower LLMs using geospatial tools for address standardization. *In*: Ku, L.-W., *et al.*, eds, *Findings of the Association for Computational Linguistics: ACL 2024.*

Association for Computational Linguistics, Bangkok, Thailand. Association for Computational Linguistics, 6048–6063.

Jelinek, F., 1990. Self-organized language modeling for speech recognition. *In*: *Readings in speech recognition*. San Francisco, CA: Morgan Kaufmann Publishers Inc., 450–506.

Kim, H., *et al.*, 2020. Natural language to SQL: where are we today? *Proceedings of the VLDB Endowment*, 13 (10), 1737–1750.

Lee, C.-H., *et al.*, 2021. KaggleDBQA: Realistic Evaluation of Text-to-SQL Parsers. *In*: *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (Volume 1: Long papers)*. Association for Computational Linguistics, 2261–2273.

Li, F., and Jagadish, H.V., 2014. Constructing an interactive natural language interface for relational databases. *Proceedings of the VLDB Endowment*, 8 (1), 73–84.

Li, H., *et al.*, 2024. Codes: Towards building open-source language models for text-to-sql. *Proceedings of the ACM on Management of Data*, 2 (3), 1–28.

Li, H., *et al.*, 2023. RESDSQL: decoupling schema linking and skeleton parsing for text-to-SQL. *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Li, Z., and Ning, H., 2023. Autonomous GIS: the next-generation AI-powered GIS. *International Journal of Digital Earth*, 16 (2), 4668–4686.

Liu, M., *et al.*, 2025. NALSpatial: a natural language interface for spatial databases. *IEEE Transactions on Knowledge and Data Engineering*, 37 (4), 2056–2070.

Molina-Villegas, A., *et al.*, 2021. Geographic named entity recognition and disambiguation in Mexican news using word embeddings. *Expert Systems with Applications*, 176, 114855.

Nelson, T.A., *et al.*, 2022. Accelerating ethics, empathy, and equity in geographic information science. *Proceedings of the National Academy of Sciences of the United States of America*, 119 (19), e2119967119.

Ning, H., *et al.*, 2025. An autonomous GIS agent framework for geospatial data retrieval. *International Journal of Digital Earth*, 18 (1), 2458688.

Smith, D.A., and Crane, G., 2001. Disambiguating geographic names in a historical digital library. *International Conference on Theory and Practice of Digital Libraries*. Springer, 127–136.

Sutskever, I., *et al.*, 2014. Sequence to sequence learning with neural networks. *In*: *Proceedings of the 28th international conference on neural information processing systems*, Volume 2, NIPS'14, Montreal, Canada. Cambridge, MA: MIT Press, 3104–3112.

Vaswani, A., 2017. Attention is all you need. *In*: I. Guyon, *et al.*, eds. *Advances in neural information processing systems*, Volume 30. Curran Associates, Inc.

Wang, B., *et al.*, 2019. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. *Annual Meeting of the Association for Computational Linguistics*.

Wu, S., *et al.*, 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv: 2303.17564*.

Yao, S., *et al.*, 2024. Lawyer GPT: A legal large language model with enhanced domain knowledge and reasoning capabilities. *In*: *Proceedings of the 2024 3rd International Symposium on Robotics, Artificial Intelligence and Information Engineering*. New York, NY: Association for Computing Machinery, 108–112.

Zhang, R., *et al.*, 2019. Editing-Based SQL Query Generation for Cross-Domain Context-Dependent Questions. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Hong Kong, China.

Zhang, Y., *et al.*, 2024a. GeoGPT: An assistant for understanding and processing geospatial tasks. *International Journal of Applied Earth Observation and Geoinformation*, 131, 103976.

Zhang, Y., *et al.*, 2024b. MapGPT: an autonomous framework for mapping by integrating large language model and cartographic tools. *Cartography and Geographic Information Science*, 51 (6), 717–743.

Zhao, X., *et al.*, 2024. Chat2data: An interactive data analysis system with rag, vector databases and llms. *Proceedings of the VLDB Endowment*, 17 (12), 4481–4484.