AtlasSQL-BR: A Brazilian Portuguese Geospatial Text-to-SQL Dataset with Spatial Hierarchies*

Diego O. Lopes[1], **Kelly R. Braghetto**[1]

[1]Institute of Mathematics, Statistics and Computer Science – University of São Paulo
São Paulo — SP — Brazil

{diego.oliveiralopes,kellyrb}@ime.usp.br

Abstract. *The Text-to-SQL task, which translates natural language into SQL queries, democratizes database access by enabling non-experts to query and analyze data effectively. While the field has seen significant progress, state-of-the-art models still fall short in processing geospatial queries due to a lack of diverse spatial data in existing training sets. This issue is compounded by the overwhelming dominance of English-language resources, leaving languages like Brazilian Portuguese largely underexplored. To bridge these gaps, we introduce AtlasSQL-BR, a novel, real-world geospatial Text-to-SQL dataset in Brazilian Portuguese. Built upon school census data and geographic boundary hierarchies, this dataset provides a robust benchmark for geospatial queries.*

1. Introduction

The capability to interact with databases is fundamental to modern information systems. However, constructing database queries requires specialized knowledge of languages such as SQL, posing a significant barrier for non-expert users. This gap between natural and query languages has driven the development of Text-to-SQL systems, which generate SQL queries directly from natural language questions (NLQs), enabling broader, democratized access to structured data [Katsogiannis-Meimarakis and Koutrika 2023].

In recent years, Large Language Models (LLMs) have substantially advanced the state of the art in Text-to-SQL tasks [Zhang et al. 2024a]. These models share the Transformer architecture [Vaswani et al. 2017], which introduced self-attention mechanisms that enable capturing long-range dependencies across both natural language and structured query tokens. Representative families include GPT [Radford and Narasimhan 2018], Llama [Grattafiori et al. 2024], and instruction-tuned variants thereof [Raiaan et al. 2024], which have achieved execution accuracies exceeding 82% on standard benchmarks [Gao et al. 2024]. Despite this progress, a critical domain remains underexplored: geospatial data.

Geospatial data encodes information about objects and events located on or near the Earth's surface [Wu et al. 2024]. Querying such data requires the use of GeoSQL, i.e., spatial SQL data types and operations generally implemented as database extensions — such as PostGIS spatial predicates (*ST_Within*, *ST_Intersects*, *ST_Buffer*) — that standard Text-to-SQL benchmarks do not cover. As a result, state-of-the-art models perform poorly

*This work was supported by São Paulo Research Foundation (FAPESP) [grants #2023/00779-0 and #2023/18026-8]; and the National Council for Scientific and Technological Development (CNPq) [grant #420623/2023-0].

on spatial queries due to the scarcity of geospatial training examples [Zhang et al. 2024b]. This limitation is compounded by the overwhelming dominance of English-language resources: languages such as Brazilian Portuguese remain largely absent from existing Text-to-SQL benchmarks, despite representing a major user population for geographic information systems used in urban planning, environmental management, and policy making.

To address these gaps, this paper introduces AtlasSQL-BR, a geospatial Text-to-SQL dataset in Brazilian Portuguese built upon real-world open-government datasets — including school census data and public facility registries — and Brazil's full official geographic boundary hierarchy: country (*país*), macro-region (*região*), state (*unidade federativa*), municipality (*município*), district (*distrito*), neighborhood (*bairro*), and street (*logradouro*). This seven-level hierarchy enables queries that require spatial reasoning across multiple administrative granularities within a single SQL statement, a challenge not addressed by any existing benchmark. The dataset comprises 980 manually annotated question-SQL pairs spanning four complexity tiers and 22 PostGIS spatial functions.

To demonstrate the utility of AtlasSQL-BR, we fine-tuned a generative LLM (Llama-3.1-8B-Instruct) on this dataset using LoRa (Low-Rank Adaptation) — a method of the Parameter-Efficient Fine-Tuning (PEFT) techniques — and evaluated both the base and fine-tuned models using a multi-metric framework. The framework combines string similarity, structural F1, component-level Jaccard, and geospatial function F1 (a metric specifically devised for geospatial function-matching), with explicit comparison between base and fine-tuned models against a gold standard.

The fine-tuned model achieved a Token F1 of 0.623, a Geospatial F1 of 0.524, and 7.69% exact spatial function match — substantially outperforming the zero-shot base model (Token F1: 0.223, Geospatial F1: 0.166) — demonstrating that LoRA adaptation meaningfully improves geospatial SQL generation even from a small manually annotated corpus and a small LLM.

2. Background and Related Work

Text-to-SQL benchmarks: Given a natural language question q and a database schema S, the Text-to-SQL task learns $f : (q, S) \rightarrow s$, where s is a syntactically valid SQL query such that $\text{Exec}(s, S) \approx \text{Ans}(q)$ [Deng et al. 2022]. Early rule-based approaches [Zelle and Mooney 1996] lacked scalability; the field evolved toward neural encoder-decoder architectures and, more recently, LLM-based approaches [Gao et al. 2024]. Since the shift to neural models, text-to-SQL has been shaped by a succession of increasingly demanding benchmarks.

WikiSQL [Zhong et al. 2017] introduced large-scale NLQ-SQL pairs but restricted queries to single-table, flat schemas. Spider [Yu et al. 2018] advanced the field with 10,181 questions across 200 cross-domain databases, requiring multi-table joins and nested queries, and established the Exact Set Match (ESM) and Execution Accuracy (EX) metrics as standard evaluation tools. BIRD [Li et al. 2023] further scaled complexity with 12,751 real-world examples and introduced the Valid Efficiency Score (VES), an integrated metric that jointly evaluates execution correctness and query efficiency, acknowledging that production systems must balance accuracy with computational cost. GeoQuery [Zelle and Mooney 1996] is the only geographically focused dataset, covering US geographic facts such as population, rivers, and state boundaries; however, its

queries are expressed as logical forms over a knowledge base rather than SQL, and rely solely on attribute comparisons without any spatial operators or hierarchical containment predicates. All these benchmarks are English-only and lack geospatial query constructs, leaving geospatial and multilingual Text-to-SQL systematically underserved.

Geospatial natural language interfaces: Geospatial databases extend SQL with spatial data types (*Point*, *LineString*, *Polygon*) and functions standardized by ISO/IEC 13249-3 [Stolze 2003]. The PostGIS extension for PostgreSQL [1] is the most widely adopted implementation, providing predicates such as *ST_Within*, *ST_Intersects*, *ST_Contains*, *ST_Buffer*, and *ST_DWithin* [Bonham-Carter 1994].

The intersection of natural language processing and geospatial databases has attracted increasing attention, driven by the difficulty of formulating spatial queries for non-expert users [Bonham-Carter 1994]. GS-SQL [Zhang et al. 2024b] is the most directly related effort to AtlasSQL-BR: it employs a geospatial semantic graph to explicitly model spatial relationships between entities, combining symbolic and continuous representations in the decoding process. GS-SQL was evaluated on Geo880, a benchmark of 880 geographic questions about US facts (states, rivers, capitals) expressed as database queries, and SpatialWiki, a dataset of spatial relationship questions derived from Wikipedia, both of which test topological and directional reasoning over geographic knowledge bases. GS-SQL demonstrated that spatial semantics require dedicated modeling beyond generic NLQ-SQL translation; however, it operates exclusively on English and does not address multi-level geographic hierarchy resolution or real-world administrative boundaries.

Liu et al. [Liu et al. 2025] proposed NALSpatial, a framework that translates natural language queries over spatial databases into executable SQL through a two-phase pipeline: a natural language understanding phase that extracts spatial entity types and query intent, and a translation phase that applies spatial entity mapping rules and structured language models. NALSpatial supports five spatial query classes — basic queries, range queries, nearest neighbor queries, spatial joins, and aggregation queries — achieving 95% translatability and 92% translation precision on controlled benchmarks. While these results are promising, NALSpatial relies on a fixed query taxonomy and pre-defined spatial entity rules, which limits its generalization to the open-ended natural language phrasing that characterizes real-world geoportal interactions and the hierarchical boundary predicates central to AtlasSQL-BR. Jiang and Yang [Jiang and Yang 2024] conducted the first study to directly evaluate LLM-based geospatial SQL generation, showing that ChatGPT can produce accurate SQL for straightforward spatial queries, including spatial joins, but struggles with complex multi-table queries and PostGIS-specific predicates — precisely the query types that dominate AtlasSQL-BR's harder complexity tiers.

Table 1 positions AtlasSQL-BR as the only dataset combining non-English language, real-world geospatial domain, PostGIS spatial operators, multi-level geographic hierarchy, and a four-tier complexity classification.

Adapting LLMs to specialized SQL domains: Several recent works have investigated prompt engineering and fine-tuning strategies for domain-specific Text-to-SQL. Nasci-

[1] PostGIS: `https://postgis.net`.

Table 1. Comparison of Text-to-SQL datasets across key dimensions.

Dataset	Lang.	Domain	Spatial	Geo Hier.	Tiers	Pairs
WikiSQL [Zhong et al. 2017]	EN	General	No	No	No	80,654
Spider [Yu et al. 2018]	EN	Cross-dom.	No	No	Yes	10,181
BIRD [Li et al. 2023]	EN	Real-world	No	No	Yes	12,751
GeoQuery [Zelle and Mooney 1996]	EN	Geography	No	No	No	880
GS-SQL [Zhang et al. 2024b]	EN	Geospatial	Yes	No	No	–
AtlasSQL-BR	**PT-BR**	**Geospatial**	**Yes**	**Yes**	**Yes (4)**	**980**

mento et al. [Nascimento et al. 2025] showed that LLMs equipped with schema-enriched prompts — specifically, LLM-friendly views that expose undocumented schemas in a structured way — achieve promising results on production-grade databases, highlighting that schema representation quality critically mediates model performance. Nan et al. [Nan et al. 2023] demonstrated through systematic prompt design experiments that few-shot prompting with carefully selected examples substantially improves semantic parsing accuracy, particularly in cross-domain settings. Coelho et al. [Coelho et al. 2024] proposed a Retrieval-Augmented Generation (RAG) approach to dynamically inject relevant schema context at inference time, bridging the gap between benchmark conditions and production database complexity. While these works collectively demonstrate the value of prompt engineering and context injection, they also expose a fundamental limitation: in domains with strict syntactic constraints — such as geospatial SQL, where a correct predicate must be selected from a closed vocabulary of spatial functions with precise geometric semantics — prompting-only strategies lack the mechanism to reliably internalize domain-specific conventions.

This motivates fine-tuning as a necessary complement: unlike prompting, supervised adaptation directly optimizes the model's parametric knowledge toward the target vocabulary and query structure, making it the appropriate strategy for AtlasSQL-BR. However, full fine-tuning of LLMs risks catastrophic forgetting when adapting to narrow domains [Kirkpatrick et al. 2017]. Low-Rank Adaptation (LoRA) [Hu et al. 2022] addresses this by injecting trainable low-rank matrices into attention layers, reducing trainable parameters by orders of magnitude with negligible performance loss. Parameter-Efficient Fine-Tuning (PEFT) has proven effective at internalizing domain-specific SQL conventions that prompting-only approaches fail to enforce consistently [Gao et al. 2024].

3. AtlasSQL-BR: Dataset Construction and Characterization

AtlasSQL-BR is motivated by two complementary gaps in the Text-to-SQL literature: the absence of geospatial benchmarks that exercise real-world spatial SQL constructs, and the underrepresentation of Brazilian Portuguese in structured query resources. The dataset integrates multiple open-government data sources to maximize thematic diversity. The primary source is the Brazilian School Census (*Censo Escolar*) [2], an annually updated administrative database that combines rich thematic attributes — school infrastructure, enrollment figures, and administrative classification — with georeferenced point geometries for each institution. Complementing the school census, the dataset also incorporates

[2]Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) - Censo Escolar da Educação Básica: `https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/censo-escolar`.

georeferenced records of public social assistance facilities, including CRAS (*Centros de Referência de Assistência Social*, community-based social assistance reference centers), CREAS (*Centros de Referência Especializados de Assistência Social*, specialized centers for high-vulnerability social assistance), and *Centros POP* (*Centros de Referência Especializado para População em Situação de Rua*, specialized centers serving the homeless population), as well as public libraries — collectively providing location information for a broad range of public service facilities distributed across the Brazilian territory.

The geographic boundary layers that support spatial containment queries are sourced from the Brazilian Institute of Geography and Statistics [3], covering seven administrative levels (as mentioned in Section 1). This combination of thematic facility data and multi-level administrative boundaries enables the construction of spatially heterogeneous queries that span from national-level containment predicates down to fine-grained district and census sector boundaries within a single SQL statement.

A central design decision is the explicit encoding of the administrative boundary levels, each represented as a polygon geometry layer in the database. This multi-level structure introduces query patterns that require models to reason about spatial containment across different administrative granularities within a single SQL statement — for instance, locating CRAS facilities within a given census sector that itself falls inside a specific municipality — a challenge absent from all existing Text-to-SQL benchmarks. Each question-SQL pair in the dataset is annotated with the relevant *territorial division* and the required *geospatial function*, enabling controlled evaluation of spatial reasoning at different levels of geographic granularity.

3.1. Relational and Spatial Schema

The database schema underlying AtlasSQL-BR is organized around the thematic facility tables and their associated geographic boundary layers. The central thematic table is `microdados_ed_basica`, derived from the Brazilian School Census microdata, which contains hundreds of attributes per school — including infrastructure indicators such as the presence and count of bathrooms, availability of internet access, television, and computer labs, as well as administrative and pedagogical attributes such as teaching modality, shift, and enrollment figures by grade level. Each school record in this table is linked to the `escola` table, which stores the school's point geometry and core identifiers including the unique identifier (`cd_entidade`), school name (`nm_entidade`), administrative dependency (`tp_depende`), and location type (`tp_localiz`). Complementing the school data, the schema also includes point geometry tables for other public facility datasets: libraries (`biblioteca`), CRAS, CREAS, and *Centros POP*, each storing thematic attributes alongside a PostGIS `geometry` column encoding the facility's geographic position as a point coordinate. The seven IBGE administrative boundary tables — from country down to census sector — store polygon geometries alongside their respective identifiers and classification attributes, enabling spatial containment joins at any level of the territorial hierarchy. Spatial relationships between facility points and boundary polygons are resolved via PostGIS predicates — primarily *ST_Intersects*, *ST_Within*, and *ST_Contains* — which determine whether a given facility falls within a particular administrative unit. Additional predicates such as *ST_Distance* and *ST_DWithin* support

[3]Instituto Brasileiro de Geografia e Estatística (IBGE): `https://www.ibge.gov.br`.

proximity-based queries that do not rely on boundary containment, further expanding the diversity of spatial reasoning patterns covered by the dataset.

3.2. Complexity-Tiered Manual Annotation

The dataset was constructed by manually elaborating and annotating 980 question-SQL pairs in Brazilian Portuguese. Each pair consists of a natural language question (*questão*), its corresponding SQL query, the territorial division it references, and the primary geospatial function it requires. Pairs are classified into four complexity tiers based on the structural properties of the SQL query, as summarized in Table 2, with all criteria AND-combined within each tier.

Table 2. Complexity tier definitions for AtlasSQL-BR.

Tier	Structural Criteria
Easy	$\leq$1 JOIN; no aggregations; no subqueries; direct spatial filter
Medium	2–3 JOINs; no aggregations; multiple spatial conditions
Hard	3–4 JOINs; $\geq$1 aggregation (COUNT, SUM, AVG); multiple spatial functions
Very Hard	4+ JOINs; multiple aggregations; subqueries and/or CTEs; window functions (ROW_NUMBER, RANK, DENSE_RANK); CASE WHEN; possible UNION

The spatial SQL vocabulary spans all 22 PostGIS functions supported by the dataset: *ST_Area*, *ST_Boundary*, *ST_Buffer*, *ST_Centroid*, *ST_Collect*, *ST_Contains*, *ST_ConvexHull*, *ST_Difference*, *ST_Distance*, *ST_DWithin*, *ST_Envelope*, *ST_GeomFromText*, *ST_Intersection*, *ST_Intersects*, *ST_IsEmpty*, *ST_IsValid*, *ST_Length*, *ST_PointOnSurface*, *ST_Transform*, *ST_UnaryUnion*, *ST_Union*, and *ST_Within*. Beyond spatial functions, the dataset covers 8 aggregation and statistical functions (AVG, COUNT, MAX, MIN, SUM, STDDEV, PERCENTILE_CONT), 5 window and ranking functions (ROW_NUMBER, RANK, DENSE_RANK, PERCENT_RANK, NTILE), conditional constructs (CASE WHEN, COALESCE, NULLIF), and structural operators including CTEs (WITH), LATERAL joins, FULL OUTER JOIN, UNION, and HAVING. Each pair was validated for syntactic executability against the target database instance.

Listing 1 presents a query example illustrating the spatial and structural complexity, as well as the challenges posed by multi-level territorial hierarchy navigation.

```
-- NLQ: Nos estados MS, MT, RO e AC, quais escolas com
--       auditorio estao contidas em seus municipios e
--       a ate 4 km da fronteira do Brasil?
SELECT e.cd_entidade, e.cd_uf, m.cd_mun
FROM   public.pais p
JOIN   public.unidade_federativa uf
       ON uf.sigla_uf IN ('MS','MT','RO','AC')
       AND ST_Intersects(uf.geometry, p.geometry)
JOIN   public.municipio m
       ON m.cd_uf = uf.cd_uf
       AND ST_Intersects(m.geometry, uf.geometry)
RIGHT JOIN public.escola e
       ON ST_Contains(m.geometry, e.geometry)
```

```
RIGHT JOIN public.microdados_ed_basica meb
       ON meb.cd_entidade = e.cd_entidade
       AND meb.in_auditorio = 1
WHERE  p.pais = 'Brasil'
AND    ST_DWithin(e.geometry::geography,
                  ST_Boundary(p.geometry)::geography, 4000);
```

Listing 1. Medium-tier query: schools with an auditorium within 4 km of Brazil's border, in four border states (country → state → municipality, microdata join, boundary proximity).

The progression across these four examples illustrates the core challenge of AtlasSQL-BR. At the Easy tier, a single spatial predicate (*ST_Intersects*) and one join suffice. At the Medium tier, the query chains three levels of the territorial hierarchy (country → state → municipality) while combining a boundary proximity predicate with a microdata attribute filter. At the Hard tier, a FULL OUTER JOIN across two regional subdivision levels (*rg_intermediaria* and *rg_imediata*) requires models to resolve partial hierarchy overlaps via `COALESCE` before performing spatial aggregation. At the Very Hard tier, five CTEs, a window function with `PARTITION BY`, a CROSS JOIN, and four nested spatial containment predicates spanning the full national-to-municipality hierarchy must be composed correctly — a query structure with no equivalent in any existing Text-to-SQL benchmark.

4. Parameter-Efficient Fine-Tuning for Geospatial SQL Generation

This section describes the fine-tuning methodology, including base model selection, LoRA adapter configuration, and the training protocol.

4.1. Base Model Selection and Justification

Selecting a base model requires balancing Portuguese language coverage, SQL generation proficiency, and single-GPU feasibility [Zhao et al. 2023]. We evaluated three candidates against a held-out subset of AtlasSQL-BR queries, prompting each model in a zero-shot setting without any fine-tuning, and measuring Exact Match and Execution Accuracy on the resulting SQL outputs. **Qwen2.5-7B-Instruct** [Yang et al. 2024], pre-trained on 18 trillion tokens across 30 languages, showed lower consistency on geospatial SQL patterns in our evaluation, likely reflecting limited Portuguese and spatial SQL representation in its post-training mixture. **Flan-T5-Large** [Chung et al. 2024], despite strong instruction-following generalization, is constrained by its encoder-decoder architecture: the decoder attends over a fixed-length representation of the input, limiting its ability to handle long schemas and complex spatial function sequences; at 780M parameters it also underperforms on multi-clause geospatial queries. **Llama-3.1-8B-Instruct** [Grattafiori et al. 2024], pre-trained on 15 trillion tokens with a 128K-token context window, extends its tokenizer with 28K non-English tokens (including Portuguese), and achieves performance comparable to leading proprietary models on code generation and multilingual benchmarks. It outperformed both alternatives across all AtlasSQL-BR metrics and was adopted as the base model.

4.2. LoRA Configuration and Training Protocol

To avoid the high computational costs of full fine-tuning, we employed Low-Rank Adaptation (LoRA) [Hu et al. 2022]. This technique freezes the pre-trained weights (W_0) and

represents the necessary updates through the decomposition $W = W_0 + \frac{\alpha}{r}BA$, where $B \in R^{d\times r}$ and $A \in R^{r\times k}$. We applied LoRA to the `q_proj` and `v_proj` attention layers using a rank (r) of 8, which defines the dimensionality of the low-rank matrices, and a scaling factor (α) of 32 to control the magnitude of the learned weights. With a dropout (p) of 0.1 to prevent overfitting, the resulting configuration renders fewer than 0.5% of the total parameters trainable. This approach effectively prevents catastrophic forgetting [Kirkpatrick et al. 2017], ensuring the model specializes in the target task while preserving its broad linguistic capabilities.

Training ran for 10 epochs with batch size 2 and 2-step gradient accumulation (effective batch 4), using the AdamW 8-bit optimizer [Dettmers et al. 2022, Loshchilov and Hutter 2019] with weight decay 0.01 and 612 warmup steps. BF16+TF32 mixed precision [Kalamkar et al. 2019] and gradient checkpointing [Chen et al. 2016] kept memory within the bounds of a single NVIDIA RTX A6000 (48 GB VRAM, 8-core CPU, 45 GB RAM). The best checkpoint was selected by minimum validation cross-entropy loss, tracked via Weights & Biases tool.

5. Experimental Evaluation

This section presents the evaluation framework and results for AtlasSQL-BR. We begin by justifying the metrics employed, then describe the experimental protocol and baselines, followed by a quantitative analysis of results, and close with a failure taxonomy characterizing the model's error modes.

5.1. Evaluation Metrics and Justification

The two canonical metrics — Exact Set Match (ESM) and Execution Accuracy (EX) — carry well-documented limitations when applied to geospatial SQL. ESM, originally proposed for relational Text-to-SQL evaluation [Yu et al. 2018], produces systematic false negatives for spatially equivalent formulations (e.g., *ST_Within(a,b)* vs. *ST_Contains(b,a)*), a problem that is exacerbated by the richer set of interchangeable spatial predicates available in PostGIS. EX requires a live database and introduces false positives when semantically different queries accidentally return identical rows on a specific database instance [Zhong et al. 2020]. We therefore adopt a **four-metric framework** that combines established string-level measures with three structure-aware metrics designed for geospatial SQL.

1. **String Similarity** [Deng et al. 2022]: three complementary surface-form measures computed over normalized, lower-cased SQL tokens after stripping comments and collapsing whitespace. *Exact Match* (EM) is a binary indicator equal to 1 if and only if the predicted string equals the gold string character-for-character after normalization [Zhong et al. 2020]. *SequenceMatcher Ratio* applies the longest-common-subsequence similarity defined as $2|M|/T$, where $|M|$ is the number of matching characters and T is the total number of characters in both strings. *Token F1* treats each SQL token as an element of a multiset and computes [Deng et al. 2022]: Token F1 $= (2\,|\hat{S} \cap S^*|)/(|\hat{S}| + |S^*|)$ where $\hat{S}$ and S^* are the token multisets of the predicted and gold queries, respectively, and $|\hat{S} \cap S^*| = \sum_t \min(\hat{S}(t),\, S^*(t))$ denotes the multiset intersection.

2. **Structural F1** [Katsogiannis-Meimarakis and Koutrika 2023]: precision, recall, and F1 computed over the multisets of Abstract Syntax Tree (AST) node descriptors extracted from the predicted and gold queries by a custom recursive-descent parser. Each node descriptor encodes the node type and its immediate label (e.g., function name, operator, clause keyword), making the metric insensitive to column aliases and clause reorderings [Yu et al. 2018]. Let $\hat{N}$ and N^* denote the AST node multisets of the predicted and gold queries, respectively:

$$P_{\text{struct}} = \frac{|\hat{N} \cap N^*|}{|\hat{N}|}, \qquad R_{\text{struct}} = \frac{|\hat{N} \cap N^*|}{|N^*|}, \qquad F1_{\text{struct}} = \frac{2\, P_{\text{struct}}\, R_{\text{struct}}}{P_{\text{struct}} + R_{\text{struct}}}$$

   Note that the node collection procedure retains type-and-label pairs (i.e., the top-two levels of each subtree), so two queries that produce identical node multisets under this representation are considered structurally equivalent.

3. **Component Jaccard** [Zhong et al. 2020]: per-clause Jaccard similarity computed independently for each SQL clause (SELECT, FROM, JOIN, WHERE, GROUP BY, ORDER BY, HAVING, LIMIT), then averaged over the clauses that are non-empty in the gold query. For each clause k, let $\hat{C}_k$ and C_k^* be the canonical token sets of the predicted and gold clause, respectively: $J_k = (|\hat{C}_k \cap C_k^*|)/(|\hat{C}_k \cup C_k^*|)$ and $\overline{J} = \frac{1}{|K|} \sum_{k \in K} J_k$, where $K = \{k : C_k^* \neq \emptyset\}$ is the set of clauses present in the gold query [Yu et al. 2018]. Clause boundaries and their tokens are determined from the AST rather than from regex splitting, so aggregate functions with internal commas (e.g., `ST_Distance(a,b)`) are never split incorrectly. For order-sensitive clauses (JOIN, ORDER BY), J_k is computed over token multisets rather than sets to penalize positional mismatches.

4. **Geospatial Function F1**: precision, recall, and F1 computed over the multiset of PostGIS spatial function calls (e.g., `ST_Contains`, `ST_Distance`, `ST_Intersects`) extracted from the predicted and gold queries via regular expression over the normalized token stream [Deng et al. 2022]. This metric specifically detects cases where a structurally plausible query uses the wrong spatial predicate — a failure mode invisible to purely syntactic metrics [Katsogiannis-Meimarakis and Koutrika 2023]. Let $\hat{G}$ and G^* denote the PostGIS function multisets of the predicted and gold queries, respectively:

$$P_{\text{geo}} = \frac{|\hat{G} \cap G^*|}{|\hat{G}|}, \qquad R_{\text{geo}} = \frac{|\hat{G} \cap G^*|}{|G^*|}, \qquad F1_{\text{geo}} = \frac{2\, P_{\text{geo}}\, R_{\text{geo}}}{P_{\text{geo}} + R_{\text{geo}}}$$

5.2. Train/Validation Partitioning

The 980 pairs were partitioned into training (80%, 784 samples) and validation (20%, 196 samples) using stratified sampling by complexity tier and spatial operator type, ensuring each tier is proportionally represented in both splits.

5.3. Experimental Protocol and Baselines

We evaluated two model variants on the 196-record AtlasSQL-BR validation split, comparing each independently against the gold SQL from the original dataset: (1) the

Table 3. Evaluation results: base vs. fine-tuned model on the AtlasSQL-BR validation set (195 comparable pairs, 196 total).

Metric	Base Model	Fine-tuned Model
Exact Match[1]	0.00	0.00
String Similarity[2]	0.076	0.302
Token Precision[2]	0.211	0.668
Token Recall[2]	0.416	0.617
Token F1[2]	0.223	0.623
Structural Precision[2]	0.298	0.351
Structural Recall[2]	0.223	0.308
Structural F1[2]	0.244	0.320
Component Jaccard[2]	0.212	0.233
Geospatial Precision[2]	0.267	0.593
Geospatial Recall[2]	0.120	0.470
Geospatial F1[2]	0.166	0.524
Spatial Exact Match[1]	0.00	7.69

[1] Percentage of predictions that exactly reproduce the gold query (*Exact Match*) or all gold spatial predicates (*Spatial Exact Match*); higher is better.

[2] Score in $[0, 1]$, averaged over the 195 evaluation pairs; higher is better.

base model (Llama-3.1-8B-Instruct, zero-shot, no fine-tuning), which serves as the untuned baseline; and (2) the **fine-tuned model** (LoRA-adapted Llama-3.1-8B-Instruct trained on the 784-sample training split). Both variants received the same prompt format: "`Traduza para SQL: {question}`" (i.e., "`Translate to SQL: {question}`"), without schema context injection. Inference was performed greedily (`do_sample=False`) with `max_new_tokens=512`. SQL was extracted from model outputs by: (i) stripping Markdown code fences if present; (ii) stripping leading annotation tags (e.g., `[Divisão: UF]`); (iii) matching the first substring beginning with `WITH` or `SELECT`; and (iv) discarding trailing non-SQL continuation text. Of the 196 validation records, 195 (99.5%) yielded SQL output from at least one model, with only 1 record skipped due to absence of any recognizable SQL token.

5.4. Quantitative Results

Table 3 presents the full evaluation results for both models across all metrics. The fine-tuned model outperforms the base model on every metric. Metrics reported as scores range from 0 to 1, where 1 indicates a perfect match; Exact Match and Spatial Exact Match are reported as percentages (0–100%) and represent the fraction of predictions that exactly reproduce the gold query or its spatial predicates, respectively.

The fine-tuned model achieves substantially higher Token F1 (0.623 vs. 0.223), indicating that fine-tuning induces a vocabulary that is far more aligned with the gold SQL token distribution. The Geospatial F1 improvement is the most diagnostically significant: the fine-tuned model reaches 0.524 against 0.166 for the base model, confirming that LoRA adaptation specifically improves the model's ability to select the correct PostGIS functions. The base model hallucinates spatial functions heavily — 211 false positives against only 77 true positives — while the fine-tuned model achieves 301 true positives with a more controlled hallucination rate (207 false positives).

Table 4. Per-function geospatial metrics for the fine-tuned model (functions with non-zero TP or FN only). TP: true positives; FP: false positives; FN: false negatives.

Function	TP	FP	FN	Recall	F1
ST_Centroid	52	15	13	0.800	0.788
ST_Distance	31	17	7	0.816	0.721
ST_DWithin	26	28	1	0.963	0.642
ST_Intersection	24	24	4	0.857	0.632
ST_Buffer	21	10	16	0.568	0.618
ST_Area	18	17	9	0.667	0.581
ST_Contains	36	25	82	0.305	0.402
ST_Intersects	28	23	54	0.342	0.421
ST_Within	65	28	153	0.298	0.418

Table 5. Top failure modes: base vs. fine-tuned model (counts over 196 records).

Failure Type	Base Count	Base %	Fine Count	Fine %
Missing spatial function	153	78.1	149	76.0
Wrong table	155	79.1	142	72.4
Wrong condition value	82	41.8	112	57.1
Missing JOIN	129	65.8	94	47.9
Wrong columns	100	51.0	93	47.4
Wrong aggregation	55	28.1	72	36.7
Wrong spatial function	58	29.6	68	34.7
Parse error	40	20.4	24	12.2

Table 4 shows the per-function geospatial breakdown for the fine-tuned model. Functions such as *ST_Centroid* (F1: 0.788), *ST_Distance* (F1: 0.721), and *ST_DWithin* (F1: 0.642) are well-learned, while *ST_Intersects* (F1: 0.421) and *ST_Within* (F1: 0.418) show lower recall, suggesting that the model conflates spatially related predicates.

5.5. Error Analysis and Failure Taxonomy

Table 5 presents the top failure modes for both models across the 196 validation records, ranked by frequency in the fine-tuned model. Three findings are noteworthy. First, fine-tuning halves the parse error rate (40 $\rightarrow$ 24), demonstrating that LoRA adaptation substantially improves SQL syntactic completeness. Second, the missing JOIN rate drops from 65.8% to 47.9%, indicating that the model learned to construct multi-table joins more reliably. Third, wrong condition value increases (41.8% $\rightarrow$ 57.1%): the fine-tuned model generates syntactically valid conditions more often, but populates them with incorrect literal values (e.g., wrong state codes or distance thresholds), a failure mode that only becomes visible once parse errors are resolved. This pattern is consistent with a model that has internalized the structural template of geospatial SQL but has not yet memorized the dataset's specific attribute vocabulary.

The string similarity distribution further confirms the improvement: the fine-tuned model has 65 records in the very-low range ($<$0.20) versus 179 for the base model, gains 83 records in the low range (0.20–0.49), rising from 15 to 98, 28 records in the medium

range (0.50–0.79), rising from 1 to 29, and 3 records in the high range (0.80–0.99) where the base model had none.

6. Discussion: Contributions, Limitations, and Research Implications

AtlasSQL-BR addresses a concrete and documented gap in the Text-to-SQL landscape: no existing benchmark combines non-English natural language, real-world geospatial data, PostGIS spatial operators, and multi-level territorial hierarchies. LoRA fine-tuning on the 784-sample split yields consistent improvements over the base model across all metrics, most notably in geospatial function recall (0.470 vs. 0.120). The halving of parse errors (20.4% $\rightarrow$ 12.2%) and the drop in missing JOIN failures (65.8% $\rightarrow$ 47.9%) show that even a small, manually curated corpus significantly improves SQL structural competence.

The persistent gap in *ST_Within* and *ST_Intersects* recall suggests the model partially conflates geometrically related predicates, which schema-aware prompting could mitigate. The rise in wrong condition value failures (41.8% $\rightarrow$ 57.1%) is a known artifact of parse-error reduction: finer-grained errors surface once malformed output is resolved [Gao et al. 2024].

The main limitations are the moderate dataset size (196 queries), the absence of execution-based evaluation, and the lack of schema-aware prompting. Since the dataset already spans multiple domains and the full IBGE territorial hierarchy, single-domain scope and restricted hierarchy are not limitations here. Future work should pursue dataset expansion, execution-based evaluation with a hosted PostGIS database, and schema-aware prompting [Nascimento et al. 2025, Nan et al. 2023].

7. Conclusion

This paper introduced AtlasSQL-BR, the first geospatial Text-to-SQL dataset in Brazilian Portuguese, built upon real-world school census data and Brazil's official territorial hierarchy. The dataset comprises 980 manually annotated question-SQL pairs covering four complexity tiers and 22 PostGIS spatial functions. We fine-tuned Llama-3.1-8B-Instruct on it using LoRA and evaluated the base and fine-tuned models against the gold standard with a multi-metric framework that includes a novel geospatial function F1 metric. Even on this small corpus, fine-tuning yields significant gains: Token F1 nearly triples (0.223 to 0.623), Geospatial F1 more than triples (0.166 to 0.524), parse errors are halved, and JOIN generation improves substantially — establishing a clear baseline for future work. The experimental framework, evaluation code,[4] and dataset[5] are publicly available.

Generative AI Disclosure

The authors disclose the use of Generative AI tools to assist in the linguistic revision and structural formatting during the drafting of this paper. The authors assume full and integral responsibility for the technical content.

[4]`https://github.com/datafromlopes/atlas-sql-br`

[5]`https://huggingface.co/datasets/datafromlopes/atlas-sql-br`

References

Bonham-Carter, G. F. (1994). *Geographic Information Systems for Geoscientists: Modelling with GIS*. Pergamon, 1st edition.

Chen, T., Xu, B., Zhang, C., and Guestrin, C. (2016). Training deep nets with sublinear memory cost. https://arxiv.org/abs/1604.06174.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. (2024). Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Coelho, G. M. C., Nascimento, E. R. S., Izquierdo, Y. T., García, G. M., Feijó, L., Lemos, M., Garcia, R. L. S., de Oliveira, A. R., Pinheiro, J. P., and Casanova, M. A. (2024). Improving the accuracy of Text-to-SQL tools based on large language models for real-world relational databases. In *Database and Expert Systems Applications*, pages 93–107. Springer Nature Switzerland.

Deng, N., Chen, Y., and Zhang, Y. (2022). Recent advances in text-to-SQL: A survey of what we have and what we expect. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. (2022). GPT3.int8(): 8-bit matrix multiplication for transformers at scale. https://arxiv.org/abs/2208.07339.

Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B., and Zhou, J. (2024). Text-to-SQL empowered by large language models: A benchmark evaluation. *Proceedings of the VLDB Endowment*, 17(5):1132–1145.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Fan, A., et al. (2024). The Llama 3 herd of models. https://arxiv.org/abs/2407.21783.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. https://arxiv.org/abs/2106.09685.

Jiang, Y. and Yang, C. (2024). Is ChatGPT a good geospatial data analyst? Exploring the integration of natural language into structured query language within a spatial database. *ISPRS International Journal of Geo-Information*, 13(1):26.

Kalamkar, D., Mudigere, D., Mellempudi, N., Das, D., Banerjee, K., Wang, S., Gu, B., Wang, Y., Chen, H., Kaul, B., and Fang, A. (2019). A study of BFLOAT16 for deep learning training.

Katsogiannis-Meimarakis, G. and Koutrika, G. (2023). A survey on deep learning approaches for text-to-SQL. *The VLDB Journal*, 32:905–936.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., Wang, B., Qin, B., Geng, R., Huo, N., Zhou, X., Ma, C., Li, G., Chang, K. C., Huang, F., Cheng, R., and Li, Y. (2023). Can LLM already serve as a database interface? a big bench for large-scale database grounded text-to-SQLs. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23. Curran Associates Inc.

Liu, M., Wang, X., Xu, J., Lu, H., and Tong, Y. (2025). NALSpatial: A natural language interface for spatial databases. *IEEE Transactions on Knowledge and Data Engineering*, 37(4):2056–2070.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Nan, L., Zhao, Y., Zou, W., Ri, N., Tae, J., Zhang, E., Cohan, A., and Radev, D. (2023). Enhancing text-to-SQL capabilities of large language models: A study on prompt design strategies. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14935–14956, Singapore. ACL.

Nascimento, E. R., García, G., Izquierdo, Y. T., Feijó, L., Coelho, G. M. C., de Oliveira, A. R., Lemos, M., Garcia, R. L. S., Leme, L. A. P. P., and Casanova, M. A. (2025). LLM-Based Text-to-SQL for Real-World databases. *SN Computer Science*, 6(2):130.

Radford, A. and Narasimhan, K. (2018). Improving language understanding by generative pre-training. `https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf`.

Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., Ahmad, J., Ali, M. E., and Azam, S. (2024). A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, 12:26839–26874.

Stolze, K. (2003). SQL/MM spatial: The standard to manage spatial data in a relational database system. In *BTW 2003–Datenbanksysteme für Business, Technologie und Web, Tagungsband der 10. BTW Konferenz*, pages 247–264. Gesellschaft für Informatik eV.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wu, J., Gan, W., Chao, H.-C., and Yu, P. S. (2024). Geospatial big data: Survey and challenges. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:17007–17020.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Dong, G., et al. (2024). Qwen2.5 technical report. `https://arxiv.org/abs/2412.15115`.

Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., and Radev, D. (2018). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921. ACL.

Zelle, J. M. and Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI)*.

Zhang, B., Ye, Y., Du, G., Hu, X., Li, Z., Yang, S., Liu, C. H., Zhao, R., Li, Z., and Mao, H. (2024a). Benchmarking the Text-to-SQL capability of large language models: A comprehensive evaluation. https://arxiv.org/abs/2403.02951.

Zhang, X., Xiao, F., Yan, L., and Zhang, Z. (2024b). GS-SQL: Modeling spatial semantics in spatial text-to-sql. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2023). A survey of large language models. `https://arxiv.org/abs/2303.18223`.

Zhong, R., Yu, T., and Klein, D. (2020). Semantic evaluation for text-to-SQL with distilled test suites. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4883–4893. ACL.

Zhong, V., Xiong, C., and Socher, R. (2017). Seq2SQL: Generating structured queries from natural language using reinforcement learning. https://arxiv.org/abs/1709.00103.