Benchmarking Large Language Models for Text-to-SQL in Brazilian Portuguese and English

Luís Felippe C. de Carvalho[1], **Paulo Sérgio dos S. Júnior**[1], **Hilário Tomaz Alves de Oliveira**[1]

[1]Programa de Pós-Graduação em Computação Aplicada (PPComp)
Instituto Federal do Espírito Santo (IFES)

felippelfcc@gmail.com, {paulo.junior, hilario.oliveira}@ifes.edu.br

Abstract. *This work assessed the performance of seventeen large language models, including open-source and proprietary models, on the Text-to-SQL task in both Brazilian Portuguese and English. Two schema representation strategies were considered: textual descriptions and representations using data definition language. Experimental results on the Spider dataset demonstrated the superior performance of proprietary models, particularly the Gemini-2.5-Flash-preview, as measured by the Execution Accuracy and Exact Match metrics. Among the open-source models, Qwen-2.5-Coder-14B achieved the highest performance. An error analysis of the best-performing model revealed strong proficiency in handling clauses such as SELECT and AND/OR, while considerable challenges persisted in generating more complex constructs, including GROUP BY with HAVING and set operators like UNION and INTERSECT.*

1. Introduction

Modern society has witnessed rapid growth in the production of digital data, driven by the digitalization of services and the adoption of computing technologies [Shi et al. 2024]. In this scenario, a substantial amount of available information is stored in relational databases, which play a crucial role in organizing and managing data across various sectors, including government, healthcare, finance, and education [Katsogiannis-Meimarakis and Koutrika 2023]. Effective access to this information is crucial for supporting informed decision-making, promoting transparency, and generating strategic value. However, accessing this information requires the use of a specialized language called Structured Query Language (SQL), whose technical proficiency is beyond the reach of most users. Thus, it creates significant barriers to the full utilization by professionals in various fields or the general population.

Natural language interface systems allow users to retrieve information stored in structured databases using natural language instead of complex SQL queries [Affolter et al. 2019]. The Text-to-SQL task aims to develop solutions that can automatically convert questions written in natural language into equivalent SQL queries [Kim et al. 2020]. The challenge of this type of system lies in generating a semantically equivalent and syntactically valid SQL query from a natural language question and a relational database schema, considering the ambiguities of natural language and the strict syntax of SQL, as well as challenges related to the database structure [Katsogiannis-Meimarakis and Koutrika 2023]. For instance, given a question, such as

"List the names of the cities in the state of Espírito Santo with a population greater than 50,000 inhabitants" and assuming that this information is in a table called City in a relational database, a Text-to-SQL system should generate a valid SQL query such as: *SELECT name FROM City WHERE population > 50,000 AND state = 'Espírito Santo'*.

Early research endeavors in Text-to-SQL sought to extract keywords from the user's natural language question and map them to possible SQL queries using a rule-based or heuristic approach [Katsogiannis-Meimarakis and Koutrika 2023]. Recently, a paradigm shift has occurred in the field with the use of large language models (LLMs), which have driven significant advances in the task, achieving outstanding levels of accuracy [Shi et al. 2024]. Despite these developments, most research has concentrated on English, resulting in a gap in the development of this kind of system for other languages. For Portuguese, a language spoken by approximately 260 million people, the development of Text-to-SQL solutions is crucial to increase accessibility and democratize access to data, particularly in institutional and governmental contexts [Pedroso et al. 2025].

In this work, we conducted a systematic investigation of the performance of seventeen proprietary and open-source LLMs, including Llama, Gemma, GPT-4, Gemini, Qwen, and others, in the Text-to-SQL task, considering questions formulated in both Portuguese and English. Our goal is to provide a comprehensive overview of the capabilities of these models in bilingual scenarios, as well as to understand the challenges involved in adopting LLMs trained predominantly on English data for use in Brazilian Portuguese. The following research questions guided the experiments performed:

> **Research Question 1 (RQ1):** *Which LLM and database schema representation strategy yields the best performance on the Text-to-SQL task for questions written in Brazilian Portuguese and English?*
>
> **Research Question 2 (RQ2):** *Which SQL clauses present the greatest challenges during SQL query generation?*

To address the research questions, experiments were conducted using the Spider dataset [Yu et al. 2018], and the performance of the LLMs was evaluated using exact match and execution accuracy, two widely adopted metrics for assessing the syntactic correctness and functional accuracy of generated SQL queries. Two database schema representation strategies were examined [Xue et al. 2023]: (i) the use of data definition language to define the table creation schema, providing a structural description of the database, and (ii) a textual representation of the tables designed to align more closely with the input formats typically processed by LLMs. Furthermore, an error analysis was conducted to identify which SQL clauses impose the most significant complexity, considering the level of complexity of the SQL queries presented in the Spider dataset. This study aims to advance the understanding of LLM applicability to the Text-to-SQL task, with a particular focus on Brazilian Portuguese. The source code developed is publicly available in a GitHub repository[1].

2. Related Work

Text-to-SQL has been the subject of extensive research due to its wide range of potential applications [Kim et al. 2020, Katsogiannis-Meimarakis and Koutrika 2023,

[1] `https://github.com/luisfelippe/llm-benchmark-text2sql-pt-en`

Shi et al. 2024]. Early approaches primarily relied on rule-based systems and template-based SQL generation tailored to specific scenarios. Although these methods demonstrated potential, they often required substantial manual effort. With advancements in deep learning, particularly the introduction of the Transformer architecture [Vaswani et al. 2017], new data-driven approaches have emerged. These models enable a more direct mapping between natural language questions and corresponding SQL queries, reducing the reliance on intermediate processes such as semantic parsing or handcrafted rules. Pre-trained Language Models (PLMs), such as BART [Lewis et al. 2020], have shown promising results.

More recently, PLMs have evolved into LLMs with improved performance capabilities as training data, and model architecture sizes have increased [Shi et al. 2024]. Two primary methodologies that have been the focus of recent research on LLM-based Text-to-SQL systems are prompt engineering and fine-tuning. By creating structured pipelines, prompt engineering leverages the ability of LLMs to follow instructions. Fine-tuning, on the other hand, entails additional training of an LLM using a task-specific learning paradigm on a particular Text-to-SQL dataset.

José and Cozman [José and Cozman 2021] addressed the challenge of translating questions formulated in Brazilian Portuguese into SQL, a language that remains underrepresented in the Text-to-SQL literature. The authors adapted the RAT-SQL+GAP [Wang et al. 2019] system to process questions written in Portuguese and translated the Spider dataset into Portuguese using the Google Cloud Translation API, followed by a manual review of the translated content. The results showed that training with the multilingual BART model (mBART-50) on a multilingual dataset comprising both English and Portuguese led to improved performance compared to training on Portuguese data alone. This finding supports the hypothesis that multilingual training can enhance performance in underrepresented languages. Although the model's performance in Portuguese was lower than in English, this gap was partially attributed to the reduced lexical similarity between Portuguese and SQL query terms.

Pedroso et al. [Pedroso et al. 2025] conducted a study to evaluate the performance of LLMs in Brazilian Portuguese, motivated by the growing need for data access in Portuguese-speaking environments. The authors investigated the effectiveness of seven LLMs, including both general-purpose and code-specific models, using the Spider dataset. Their contributions included translating and validating the test partition of the Spider into Portuguese, which had not been addressed in the work of José and Cozman [José and Cozman 2021]. The translation was carried out using OpenAI's GPT-4o mini model, followed by manual verification. The study employed a zero-shot setting and used exact match and execution accuracy as evaluation metrics. The results indicated that larger models and those specialized in code generation outperformed smaller and general-purpose models, with reduced performance differences between the English and Portuguese tasks. An important observation was the decreasing performance gap between the two languages as the number of model parameters increased, suggesting that larger LLMs possess a greater capacity for linguistic adaptation.

This work builds upon previous studies by advancing the investigation of the Text-to-SQL task through a comprehensive and systematic evaluation using the Spider dataset in both Brazilian Portuguese and English. Specifically, it expands the evaluation to include seventeen LLMs with diverse architectures, including proprietary models, thereby increasing the breadth of systems assessed. Additionally, it provides a comparative analysis of two database schema representation strategies: textual descriptions and representations based on data definition language (DDL). The evaluation, conducted using the Exact Match and Execution Accuracy metrics, was further stratified according to the SQL query difficulty levels defined in the Spider dataset. An error analysis was also performed by examining Exact Match at the level of individual SQL query components, yielding insights into the specific challenges faced in generating complex clauses. To the best of our knowledge, such analyses have not been previously conducted in studies addressing the Brazilian Portuguese language.

3. Method

This section details the experimental methodology used in this work. Initially, the Spider dataset is presented, followed by a description of the selected LLMs and their key features. Finally, the experimental setup is presented, focusing on the two strategies for representing the database tables used to create the prompt, the experiments performed, and the evaluation measures used to analyze the models' performance.

3.1. Spider Dataset

The experiments were performed using the Spider database, a widely employed benchmark for the Text-to-SQL task [Yu et al. 2018]. This dataset was originally developed in English and comprises 10,181 questions and 5,693 SQL queries, distributed across 200 databases from 138 different domains. Each database record includes a natural language question, the corresponding SQL query that answers the question, the database schema, and the degree of complexity of the SQL query. The level of complexity is defined based on the number of components, selections, and conditions that the SQL query incorporates, such as GROUP BY and ORDER BY, set operators such as INTERSECT and EXCEPT, and nested subqueries. The classification includes four levels (easy, medium, hard, and extra hard) based on specific thresholds of combinations of these elements [Yu et al. 2018]. This systematic stratification enables a granular analysis of model performance in the face of the intrinsic complexity of the queries, allowing the identification of specific challenges inherent to the task.

The original database is divided into three partitions: a training partition with 7,000 samples, a development partition with 1,034 samples, and a testing partition with 2,147 samples. In this work, we used the translated version of the Spider database developed by José and Cozman [José and Cozman 2021], which was translated from the training and development sets. It is important to note that this work employed only the validation set, which comprises 2,068 samples, consisting of 1,034 in English and 1,034 in Portuguese. Table 1 presents examples extracted from the Spider database, with one record provided for each complexity level.

3.2. Investigated LLMs

The experiments carried out included a diverse set of LLMs to evaluate their performance in the Text-to-SQL task. The selection of models was divided into three categories, aiming to cover a representative spectrum of architectures, capabilities, and accessibility.

Table 1. Examples of records extracted from the Spider dataset.

Question (EN)	Question (PT-BR)	SQL Query	Complexity
What are all distinct countries where singers above age 20 are from?	Quais são os países distintos de onde vêm os cantores com mais de 20 anos?	SELECT DISTINCT country FROM singer WHERE age >20	Easy
What are the names, countries, and ages for every singer in descending order of age?	Mostre o nome, o país e a idade de todos os cantores, ordenados por idade, do mais velho ao mais novo.	SELECT name, country, age FROM singer ORDER BY age DESC	Medium
List all song names by singers above the average age.	Liste todos os nomes de músicas por cantores acima da idade média.	SELECT song_name FROM singer WHERE age >(SELECT avg(age) FROM singer)	Hard
Show the stadium name and capacity with most number of concerts in year 2014 or after.	Mostre o nome e a capacidade do estádio com o maior número de shows no ano de 2014 ou depois.	SELECT T2.name, T2.capacity FROM concert AS T1 JOIN stadium AS T2 ON T1.stadium_id = T2.stadium_id WHERE T1.year >= 2014 GROUP BY T2.stadium_id ORDER BY count(*) DESC LIMIT 1	Extra Hard

The variation in the number of parameters and the specializations of the selected models allows for a varied analysis of the capacity of different architectures to handle the complexity inherent in the task.

The first category comprises open-source models, selected with the aim of analyzing the performance of solutions that do not require high-performance infrastructures. This group included the Gemma 3 models [Team et al. 2025], with variations of 1B, 4B and 12B billion parameters; Llama 3.1 [Grattafiori et al. 2024], with 8 billion (8B) parameters; Llama 3.2, with variations of 1B and 3B billion parameters; Llama 3.3, with 70 billion (70B) parameters; and Qwen 2.5 [Yang et al. 2025], with variations of 3B and 7B billion parameters.

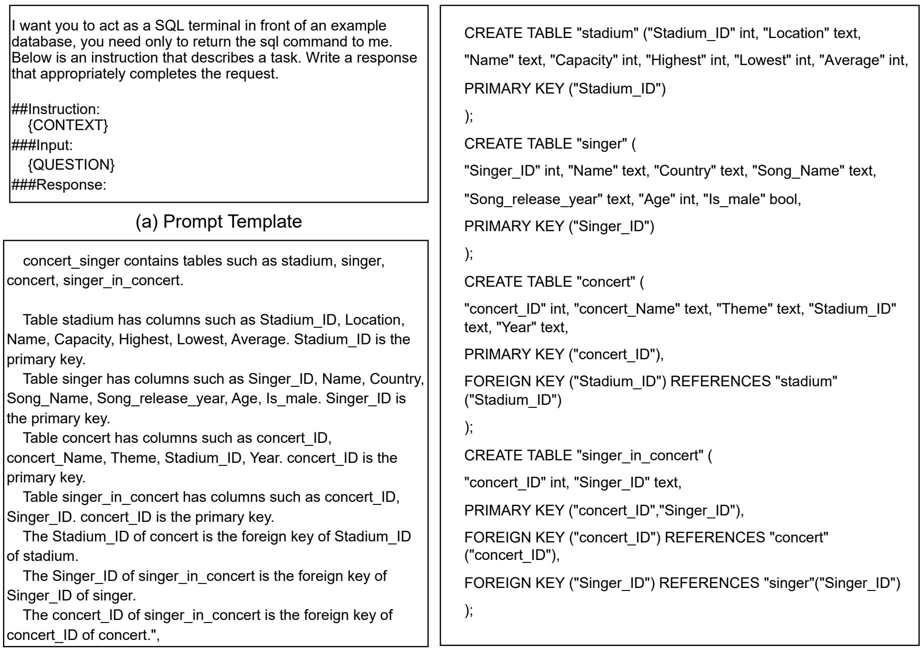
The second category includes models specialized in code generation and comprehension tasks, which are also open-source and have varying numbers of parameters. The goal of this inclusion was to comparatively analyze the performance of generalist architectures in relation to those specially optimized for programming tasks. The selected models were CodeGemma [Team et al. 2024]; CodeLlama [Roziere et al. 2023], both with 7 billion (7B) parameters; and Qwen 2.5 Coder [Hui et al. 2024], with 14 billion (14B) parameters.

Finally, the third category comprises commercially available proprietary models, accessible through application programming interfaces (APIs) from the respective companies. The inclusion of these models aims to establish a comparison with the cutting-edge solutions currently available, serving as high-performance baselines. The models evaluated were OpenAI's GPT-4o [Hurst et al. 2024]; Anthropic's Claude 3.7 Sonnet [Anthropic 2025]; Gemini 2.5 Pro preview-05-06 [Google 2025c], Gemini 2.5 Flash preview-04-17 [Google 2025b] and Google's Gemini 2.0 Flash [Google 2025a].

3.3. Experimental Setup

To address the two research questions (RQ1 and RQ2), two experiments were conducted. The first experiment aimed to evaluate the impact of two schema representation strategies on the performance of the LLMs, considering both languages (English and Portuguese) and the complexity level of the target SQL queries. This evaluation employed the execution accuracy (EX) metric. The second experiment involved an error analysis of individual components of the SQL queries. For this purpose, the F1-score was computed based on the exact match (EM) metric.

To answer RQ1, we investigate the impact of the database schema representation on the performance of the models for the Text-to-SQL task. Two distinct representation strategies were defined based on Xue et al. [Xue et al. 2023]. Thus, the zero-shot prompt provided to the model follows the composition of the prompt template (Figure 1a), enriched by the selected context and the respective question in natural language. In the first approach (Figure 1b), each dataset sample was enriched with a natural language description of the database tables relevant to the given question. On the other hand, the second version of the dataset associated each question with the data definition language (DDL) statements corresponding to the relevant database tables, as shown in Figure 1c.



```
I want you to act as a SQL terminal in front of an example
database, you need only to return the sql command to me.
Below is an instruction that describes a task. Write a response
that appropriately completes the request.

##Instruction:
   {CONTEXT}
###Input:
   {QUESTION}
###Response:
```

(a) Prompt Template

concert_singer contains tables such as stadium, singer, concert, singer_in_concert.

Table stadium has columns such as Stadium_ID, Location, Name, Capacity, Highest, Lowest, Average. Stadium_ID is the primary key.
Table singer has columns such as Singer_ID, Name, Country, Song_Name, Song_release_year, Age, Is_male. Singer_ID is the primary key.
Table concert has columns such as concert_ID, concert_Name, Theme, Stadium_ID, Year. concert_ID is the primary key.
Table singer_in_concert has columns such as concert_ID, Singer_ID. concert_ID is the primary key.
The Stadium_ID of concert is the foreign key of Stadium_ID of stadium.
The Singer_ID of singer_in_concert is the foreign key of Singer_ID of singer.
The concert_ID of singer_in_concert is the foreign key of concert_ID of concert.",

(b) Textual Context

```
CREATE TABLE "stadium" ("Stadium_ID" int, "Location" text,
"Name" text, "Capacity" int, "Highest" int, "Lowest" int, "Average" int,
PRIMARY KEY ("Stadium_ID")
);
CREATE TABLE "singer" (
"Singer_ID" int, "Name" text, "Country" text, "Song_Name" text,
"Song_release_year" text, "Age" int, "Is_male" bool,
PRIMARY KEY ("Singer_ID")
);
CREATE TABLE "concert" (
"concert_ID" int, "concert_Name" text, "Theme" text, "Stadium_ID"
text, "Year" text,
PRIMARY KEY ("concert_ID"),
FOREIGN KEY ("Stadium_ID") REFERENCES "stadium"
("Stadium_ID")
);
CREATE TABLE "singer_in_concert" (
"concert_ID" int, "Singer_ID" text,
PRIMARY KEY ("concert_ID","Singer_ID"),
FOREIGN KEY ("concert_ID") REFERENCES "concert"
("concert_ID"),
FOREIGN KEY ("Singer_ID") REFERENCES "singer"("Singer_ID")
);
```

(c) DDL Context

Figure 1. Prompt template and schema representation strategies investigated.

To evaluate the results, the official Spider metrics served as the basis for the quantitative evaluation. The EM metric evaluates the equivalence between a predicted SQL query and a reference SQL query. This metric determines the exact match of all structural components, such as clauses and keywords. The evaluation treats each clause as a set of subcomponents, mitigating order sensitivity and focusing on the structural correctness and global semantics of the query. The EX metric assesses the ability of an automatically generated SQL query to be executable and yield the same results as a reference query. The EX metric is computed by executing the reference query and the automatically generated query in the SQLite database.

A set of libraries was used to operationalize the experiments with the open-source models. Model loading and optimization for inference were facilitated by the Unsloth library[2], known for its ability to accelerate the training and execution of LLMs, especially on hardware with limited resources. Unsloth was used in conjunction with the Transformers library, developed by Hugging Face[3], which provided the fundamental interface for accessing and manipulating the various open-source model architectures. The orchestration of model execution, including the formulation of prompts, sending requests to various LLMs (both local open-source models and proprietary models accessed via API), and collecting generated responses, was managed through the LangChain framework[4].

4. Results

This section presents the results obtained in the performed experiments. The analysis focuses on the performance of various LLMs under different contextualization conditions, including the database schema and question language. The results are organized into two subsections: the first details the performance evaluation of the LLMs, considering the execution accuracy (EX) metric, and the second focuses on the quantitative analysis of errors using the exact match (EM) metric.

4.1. RQ 1 - LLMs Assessment

Table 2 and Table 3 present the results of this first experiment, considering textual representations and using DDL, respectively. The performance of the models was measured using the EX metric, considering questions in English (EN) and Brazilian Portuguese (PT-BR), segmented by the reference SQL query difficulty level (EASY, MEDIUM, HARD, EXTRA), and aggregated (ALL). The best results are highlighted in bold.

Table 2. Experimental results based on Execution Accuracy using the textual representation.

	EN					PT-BR				
LLMs	**EASY**	**MEDIUM**	**HARD**	**EXTRA**	**ALL**	**EASY**	**MEDIUM**	**HARD**	**EXTRA**	**ALL**
Claude 3.7 Sonnet	0.79	0.34	0.16	0.05	0.37	0.75	0.33	0.17	0.06	0.36
CodeGemma-7b	0.48	0.23	0.11	0.10	0.25	0.35	0.13	0.06	0.07	0.16
CodeLlama-7b	0.39	0.33	0.25	0.11	0.30	0.35	0.24	0.20	0.05	0.23
Gemini-2.0-Flash	0.71	0.52	0.64	0.42	0.57	0.72	0.48	0.49	0.40	0.53
Gemini-2.5-Flash-preview	**0.86**	**0.74**	**0.70**	**0.39**	**0.73**	**0.84**	**0.68**	**0.63**	**0.51**	**0.68**
Gemini-2.5-pro-preview	0.67	0.63	0.57	0.45	0.60	0.71	0.60	0.53	0.40	0.58
Gemma-3-1b	0.15	0.07	0.02	0.01	0.07	0.08	0.04	0.01	0.00	0.04
Gemma-3-4b	0.58	0.34	0.33	0.30	0.39	0.56	0.29	0.25	0.13	0.32
Gemma-3-12b	0.71	0.48	0.38	0.30	0.49	0.65	0.44	0.39	0.28	0.45
GPT-4o	0.44	0.39	0.29	0.17	0.35	0.40	0.33	0.25	0.11	0.30
Llama-3.2-1B	0.16	0.06	0.03	0.00	0.07	0.03	0.03	0.02	0.00	0.02
Llama-3.2-3B	0.58	0.31	0.17	0.10	0.32	0.45	0.20	0.12	0.06	0.22
Llama-3.1-8B	0.34	0.22	0.11	0.11	0.22	0.26	0.17	0.12	0.08	0.17
Llama-3.3-70B	0.60	0.40	0.42	0.15	0.41	0.59	0.35	0.37	0.13	0.38
Qwen2.5-3B	0.38	0.28	0.21	0.13	0.27	0.29	0.22	0.16	0.09	0.21
Qwen2.5-7B	0.77	0.45	0.37	0.18	0.47	0.64	0.29	0.31	0.16	0.36
Qwen2.5-Coder-14B	0.87	0.63	0.47	0.40	0.63	0.82	0.56	0.44	0.36	0.57

The Gemini family of models, particularly Gemini-2.5-Flash-preview, attained the highest EX scores across all evaluated configurations: 0.73 for English and 0.68 for Portuguese using textual representations, and 0.66 for English and 0.61 for Portuguese with DDL-based representations. This model consistently outperformed the others, followed

[2] https://unsloth.ai/

[3] https://huggingface.co/

[4] https://www.langchain.com/

by the Gemini-2.5-Pro-preview. Among the open-source models, Qwen2.5-Coder-14B demonstrated the best performance.

The comparison between English and Portuguese revealed a trend of higher accuracy for the original data in English in most scenarios, suggesting greater robustness for this language due to the predominant training data available for the English language. The influence of the type of contextualization of the database schema (textual description vs. DDL code) varied between models. For Gemini-2.5-Flash-preview, textual contextualization resulted in higher accuracy in both languages. In contrast, GPT-4o demonstrated an increase in performance with DDL contextualization, both in English and Portuguese.

The specialized model Qwen2.5-Coder-14B exhibited good performance, presenting stability in English with both contextualizations, achieving a value of 0.63 in English across all levels of complexity and showing minimal differences in Portuguese. This result corroborates the hypothesis that architectures optimized for programming tasks can offer advantages, consistently positioning it as the most performant among open-source models, outperforming even generalist models with a greater number of parameters. Regarding the open-source models (Gemma 3, Llama, Qwen 2.5), a positive correlation was observed between the increase in the number of parameters and accuracy, with smaller models presenting the lowest accuracy rates.

Table 3. Experimental results based on Execution Accuracy using the DDL representation.

	EN					PT-BR				
LLMs	**EASY**	**MEDIUM**	**HARD**	**EXTRA**	**ALL**	**EASY**	**MEDIUM**	**HARD**	**EXTRA**	**ALL**
Claude 3.7 Sonnet	0.75	0.36	0.17	0.06	0.38	0.73	0.32	0.16	0.06	0.35
CodeGemma-7b	0.54	0.26	0.09	0.10	0.27	0.37	0.19	0.05	0.07	0.19
CodeLlama-7b	0.33	0.14	0.08	0.10	0.17	0.31	0.12	0.11	0.05	0.15
Gemini-2.0-Flash	0.70	0.56	0.47	0.31	0.53	0.69	0.53	0.42	0.37	0.52
Gemini-2.5-Flash-preview	**0.83**	**0.70**	**0.57**	**0.40**	**0.66**	**0.80**	**0.64**	**0.52**	**0.32**	**0.61**
Gemini-2.5-Pro-preview	0.67	0.60	0.47	0.42	0.56	0.66	0.57	0.45	0.42	0.55
Gemma-3-1b	0.17	0.05	0.03	0.01	0.07	0.11	0.02	0.01	0.00	0.03
Gemma-3-4b	0.62	0.42	0.40	0.19	0.43	0.58	0.35	0.30	0.16	0.37
Gemma-3-12b	0.75	0.44	0.40	0.17	0.47	0.69	0.43	0.37	0.20	0.45
GPT-4o	0.58	0.43	0.33	0.18	0.41	0.48	0.37	0.24	0.13	0.34
Llama-3.2-1B	0.17	0.08	0.09	0.01	0.09	0.03	0.02	0.02	0.00	0.02
Llama-3.2-3B	0.50	0.28	0.22	0.07	0.29	0.48	0.20	0.16	0.06	0.24
Llama-3.1-8B	0.25	0.21	0.16	0.07	0.19	0.32	0.17	0.11	0.06	0.18
Llama-3.3-70B	0.57	0.38	0.38	0.13	0.39	0.54	0.33	0.31	0.11	0.34
Qwen2.5-3B	0.50	0.37	0.27	0.14	0.35	0.44	0.30	0.25	0.09	0.29
Qwen2.5-7B	0.73	0.47	0.44	0.19	0.48	0.68	0.42	0.39	0.22	0.45
Qwen2.5-Coder-14B	0.87	0.63	0.49	0.40	0.63	0.82	0.56	0.42	0.36	0.56

All LLMs exhibited a performance drop as question complexity increased *(EASY $>$ MEDIUM $>$ HARD $>$ EXTRA)*. More robust models, such as Gemini-2.5-Flash-preview and Qwen2.5-Coder-14B, maintained more stable performance at higher difficulty levels (HARD and EXTRA). The EXTRA category presented a substantial challenge, with considerable decreases in accuracy, indicating a limitation in generalization ability for high-complexity queries. Claude 3.7 Sonnet demonstrated high performance on low complexity questions, but its accuracy dropped sharply at higher difficulty levels.

Overall, the results indicate that Gemini-2.5-Flash-preview consistently achieved the highest performance across the evaluated settings. Among the open-source models, Qwen2.5-Coder-14B proved to be an effective alternative. The language of the input questions and the database schema representation strategy exhibited varying impacts on performance depending on the model architecture, underscoring the importance of optimization for specific linguistic and structural contexts. The textual schema representation consistently outperformed the DDL-based approach, and models achieved higher performance on English questions compared to those in Portuguese.

4.2. RQ 2 - Error Analysis

To further understand the types of errors considering both representations (textual and DDL), a performance analysis was conducted using the F1-score, computed based on the EM metric at the component level of the generated SQL queries. Due to space constraints, this analysis focused on the Gemini 2.5 Flash-preview, as it demonstrated the best overall performance in the first experiment. The metrics were computed for the main SQL clauses: *SELECT* with and without aggregate functions (AGG), *WHERE* with and without explicit operators, *GROUP BY* with and without *HAVING*, *ORDER BY*, logical operators (*AND/OR*), set operators (*IUEN - Intersect, Union, Except, None*), and other keywords *keywords*. Table 4 presents the results, distinguishing between the performance for English (EN) and Portuguese (PT-BR), segmented by difficulty level and representation strategy.

Table 4. Performance of the Gemini 2.5 Flash-preview in the F1-score of the EM metric with textual and DDL contextualization.

	Textual Representation									
SQL Clauses	**EN**					**PT**				
	EASY	**MEDIUM**	**HARD**	**EXTRA**	**ALL**	**EASY**	**MEDIUM**	**HARD**	**EXTRA**	**ALL**
Select	0.93	0.86	0.90	0.79	0.88	0.94	0.85	0.86	0.75	0.86
Select (no AGG)	0.94	0.88	0.90	0.79	0.88	0.94	0.86	0.86	0.75	0.86
Where	0.89	0.82	0.67	0.54	0.76	0.92	0.82	0.64	0.54	0.76
Where (no OP)	0.91	0.82	0.75	0.63	0.79	0.94	0.83	0.69	0.63	0.79
Group (no Having)	0.92	0.80	0.91	0.76	0.81	0.95	0.78	0.87	0.73	0.79
Group	0.72	0.74	0.83	0.70	0.74	0.84	0.70	0.81	0.69	0.72
Order	0.89	0.95	0.91	0.59	0.82	0.82	0.93	0.80	0.65	0.80
And/Or	1.00	0.99	0.98	0.97	**0.99**	1.00	0.99	0.97	0.96	**0.98**
IUEN	1.00	1.00	0.49	0.55	0.51	1.00	1.00	0.40	0.31	0.36
keywords	0.92	0.91	0.76	0.76	0.86	0.93	0.91	0.72	0.76	0.85
	DDL Rrepresentation									
	EN					**PT**				
	EASY	**MEDIUM**	**HARD**	**EXTRA**	**ALL**	**EASY**	**MEDIUM**	**HARD**	**EXTRA**	**ALL**
Select	0.92	0.85	0.81	0.70	0.84	0.93	0.81	0.79	0.63	0.81
Select (no AGG)	0.92	0.86	0.81	0.70	0.84	0.93	0.81	0.79	0.63	0.82
Where	0.90	0.80	0.59	0.46	0.73	0.91	0.73	0.57	0.42	0.70
Where (no OP)	0.92	0.81	0.65	0.50	0.76	0.92	0.75	0.61	0.48	0.72
Group (no Having)	0.83	0.74	0.85	0.68	0.78	0.86	0.76	0.89	0.54	0.73
Group	0.72	0.70	0.83	0.65	0.68	0.74	0.68	0.86	0.51	0.67
Order	0.96	0.93	0.86	0.52	0.78	0.86	0.89	0.77	0.48	0.74
And/Or	1.00	0.99	0.97	0.95	**0.98**	1.00	0.99	0.96	0.95	**0.98**
IUEN	1.00	1.00	0.53	0.46	0.50	1.00	1.00	0.38	0.34	0.36
keywords	0.91	0.87	0.68	0.65	0.80	0.93	0.83	0.65	0.57	0.77

The analysis of aggregated F1-scores (ALL) revealed varying performance across different SQL clauses. Logical operators such as *AND* and *OR* exhibited high accuracy, with values approaching 100 percent (0.99 for English and 0.98 for Portuguese in the textual context; and 0.98 for both languages in the DDL context), indicating the models' strong ability to combine conditions correctly. The *SELECT* clause, including its variant without aggregation functions (*SELECT* without *AGG*), also achieved robust results, ranging from 0.81 to 0.88, depending on the context and language.

Other clauses exhibited moderate performance, along with specific challenges. The *KEYWORDS* category (such as *DISTINCT* and *LIMIT*) obtained F1-scores between 0.77 and 0.86, suggesting some difficulty in using these modifiers. The *ORDER BY* clause registered scores from 0.74 to 0.82, with performance declining in more complex scenarios (*EXTRA*). Similarly, the *WHERE* clause presented aggregate results between 0.70 and 0.76, with some of the errors being in choosing the correct operator. The most significant difficulties were observed in more complex clauses. The F1-score for *GROUP BY* (including *HAVING*) ranged from 0.67 to 0.74, with the correct generation of the *HAVING* clause being a critical error point. Set operators (*IUEN*) consistently performed the worst, with low aggregate scores (0.36 to 0.51), especially at *HARD* and *EXTRA* difficulty levels.

Comparison between contextualization strategies (textual *vs.* DDL) indicated that textual contextualization resulted in slightly higher F1-scores for most clauses, suggesting that textual description of the schema may be marginally more beneficial for the correct formation of SQL components. Regarding language, performance in English was generally superior to that in Portuguese, particularly for the *IUEN* clause, which may reflect a bias in the training data. Additionally, the F1-score for almost all clauses decreased as question complexity increased.

5. Conclusions

This work evaluated the performance of seventeen large language models (LLMs), including both generalist and specialized code architectures, as well as open-source and proprietary models, in the task of translating natural language questions written in Brazilian Portuguese and English into SQL queries (Text-to-SQL). Using the Spider benchmark and a zero-shot approach, two database schema representation strategies were tested: textual description and DDL code. Performance was measured by Exact Match and Execution Accuracy metrics, segmented by the level of complexity of the SQL queries and the language used.

The results of the experiments indicated that proprietary models, such as Gemini 2.5 Flash-preview, obtained the highest accuracy, while the Qwen 2.5 Coder-14B stood out among the open-source models. A trend of improved performance was observed for English questions, with a variable influence on the type of schema representation across the different models. The analysis of the errors revealed persistent challenges in generating more complex SQL clauses, such as *GROUP BY* with *HAVING* and set operators (*IUEN*), especially for questions written in Portuguese. Finally, it was observed that all models exhibited a high error rate in correctly generating SQL queries for the more complex queries categorized as *HARD* and *EXTRA*, especially in situations involving JOIN clauses and subqueries.

To further enhance model performance and support the development of a real-world Text-to-SQL application, future research directions include fine-tuning open-source models, evaluating LLM performance on real-world datasets, exploring advanced prompting strategies, and designing post-processing mechanisms to correct errors in the generated SQL queries, including regeneration steps based on identified mistakes.

Acknowledgements

This research is funded in part by FAPES (993/2023 P: 2023-N3HJD).

References

Affolter, K., Stockinger, K., and Bernstein, A. (2019). A comparative survey of recent natural language interfaces for databases. *The VLDB Journal*, 28(5):793–819.

Anthropic (2025). Claude 3.7 sonnet system card. `https://assets.anthropic.com/m/785e231869ea8b3b/original/claude-3-7-sonnet-system-card.pdf`. Acessado em: (27/05/2025).

Google, I. (2025a). Gemini 2.0 flash. `https://storage.googleapis.com/model-cards/documents/gemini-2-flash-lite.pdf`. Acessado em: (27/05/2025).

Google, I. (2025b). Gemini 2.5 flash preview. `https://storage.googleapis.com/model-cards/documents/gemini-2.5-flash-preview.pdf`. Acessado em: (27/05/2025).

Google, I. (2025c). Gemini 2.5 pro preview. `https://storage.googleapis.com/model-cards/documents/gemini-2.5-pro-preview.pdf`. Acessado em: (27/05/2025).

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hui, B., Yang, J., Cui, Z., Yang, J., Liu, D., Zhang, L., Liu, T., Zhang, J., Yu, B., Lu, K., et al. (2024). Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. (2024). Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

José, M. A. and Cozman, F. G. (2021). mrat-sql+ gap: a portuguese text-to-sql transformer. In *Intelligent Systems: 10th Brazilian Conference, BRACIS 2021, Virtual Event, November 29–December 3, 2021, Proceedings, Part II 10*, pages 511–525. Springer.

Katsogiannis-Meimarakis, G. and Koutrika, G. (2023). A survey on deep learning approaches for text-to-sql. *The VLDB Journal*, 32(4):905–936.

Kim, H., So, B.-H., Han, W.-S., and Lee, H. (2020). Natural language to sql: Where are we today? *Proceedings of the VLDB Endowment*, 13(10):1737–1750.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Pedroso, B., Pereira, M., and Pereira, D. (2025). Performance evaluation of llms in the text-to-sql task in portuguese. In *Anais do XXI Simpósio Brasileiro de Sistemas de Informação*, pages 260–269, Porto Alegre, RS, Brasil. SBC.

Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., Adi, Y., Liu, J., Sauvestre, R., Remez, T., et al. (2023). Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Shi, L., Tang, Z., Zhang, N., Zhang, X., and Yang, Z. (2024). A survey on employing large language models for text-to-sql tasks. *ACM Computing Surveys*.

Team, C., Zhao, H., Hui, J., Howland, J., Nguyen, N., Zuo, S., Hu, A., Choquette-Choo, C. A., Shen, J., Kelley, J., et al. (2024). Codegemma: Open code models based on gemma. *arXiv preprint arXiv:2406.11409*.

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al. (2025). Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, B., Shin, R., Liu, X., Polozov, O., and Richardson, M. (2019). Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. *arXiv preprint arXiv:1911.04942*.

Xue, S., Jiang, C., Shi, W., Cheng, F., Chen, K., Yang, H., Zhang, Z., He, J., Zhang, H., Wei, G., et al. (2023). Db-gpt: Empowering database interactions with private large language models. *arXiv preprint arXiv:2312.17449*.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. (2025). Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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 Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.