Performance evaluation of LLMs in the Text-to-SQL task in Portuguese

Breno Carvalho Pedroso
breno.cpedroso@gmail.com
Universidade Federal de Lavras
Lavras, MG, Brazil

Marluce Rodrigues Pereira
marluce@ufla.br
Universidade Federal de Lavras
Lavras, MG, Brazil

Denilson Alves Pereira
denilsonpereira@ufla.br
Universidade Federal de Lavras
Lavras, MG, Brazil

Abstract

Context: The rising need for consulting data in industry and academic contexts has fueled Text-to-SQL development, where natural language queries are translated into SQL, making data access easier. **Problem:** Most research focuses on English Text-to-SQL, leaving Portuguese—a language spoken by over 260 million people—underrepresented, creating challenges for organizations reliant on accurate data retrieval in Portuguese. **Solution:** This study evaluates the effectiveness of various Large Language Models (LLMs) on Portuguese Text-to-SQL tasks using a validated translation of the Spider benchmark. **IS Theory:** This research applies Task-Technology Fit (TTF) Theory to assess how well LLMs meet the needs of Portuguese Text-to-SQL tasks. TTF evaluates the match between LLM capabilities and task requirements, including language understanding, schema recognition, and SQL generation. Proper alignment is key for effective data retrieval, particularly for Portuguese-language applications in organizational decision-making. **Method:** A comparative analysis of seven LLMs—tested on both Portuguese and English Spider benchmarks—was performed. Exact Match (EM) and Execution Accuracy (EX) metrics measured performance, and a zero-shot prompting approach maintained consistency. **Results Summary:** Larger LLMs and specialized code models excelled, showing less performance variance between Portuguese and English tasks. Generalist models, however, produced verbose outputs, which may limit practical use in production systems. **Contributions and Impact on IS:** This research establishes baseline Portuguese Text-to-SQL metrics and insights into language adaptability in LLMs, offering guidance for organizations seeking Portuguese-language data solutions. By bridging language gaps, it advances data-driven practices and fosters growth in Portuguese-speaking regions.

CCS Concepts

• **Information systems** → **Data management systems**; • **Computing methodologies** → *Natural language processing*; • **Software and its engineering** → Software creation and management.

Keywords

Text-to-SQL, Large Language Models, Database Querying, Code Generation, Cross-Lingual Transfer, Spider Dataset

1 INTRODUÇÃO

A crescente demanda por acesso a dados na indústria e na academia tem impulsionado o desenvolvimento de novos meios para interação com bancos de dados. Neste contexto, a tarefa de *Text-to-SQL* emerge como uma solução promissora, permitindo que usuários formulem consultas em linguagem natural que são automaticamente convertidas em comandos na linguagem SQL [6, 22, 31]. Essa abordagem democratiza o acesso a dados estruturados, possibilitando que profissionais sem conhecimento técnico em SQL possam realizar consultas complexas de maneira eficiente.

Com o advento dos *Large Language Models* (LLMs), a tarefa de *Text-to-SQL* experimentou avanços significativos, alcançando níveis de acurácia anteriormente considerados inatingíveis. Modelos como GPT-4 demonstraram capacidade de compreender nuances linguísticas e gerar consultas SQL precisas a partir de instruções em linguagem natural [6]. No entanto, assim como em outras aplicações de processamento de linguagem natural, a maioria das pesquisas e desenvolvimentos nesta área tem se concentrado primariamente no idioma inglês [3], criando uma lacuna significativa para usuários de outras línguas.

Para a comunidade lusófona, que engloba mais de 260 milhões de falantes em nove países, a disponibilidade de soluções *Text-to-SQL* em português representa não apenas uma questão de acessibilidade, mas também um imperativo para a democratização do acesso a dados. A necessidade de interagir com bancos de dados em língua portuguesa é particularmente relevante em contextos corporativos e governamentais, onde a precisão na interpretação das consultas é crucial para a tomada de decisões.

Sob a perspectiva teórica de Sistemas de Informação, este estudo se fundamenta na teoria de Task-Technology Fit (TTF), que avalia o grau de alinhamento entre as capacidades de uma tecnologia e os requisitos das tarefas que ela deve suportar [8]. No contexto de *Text-to-SQL* em português, o TTF oferece um framework valioso para analisar como os LLMs atendem às necessidades específicas de consultas em língua portuguesa. Essa teoria é particularmente relevante pois permite avaliar sistematicamente a adequação dos modelos em diferentes dimensões: compreensão linguística do português, reconhecimento do esquema do banco de dados e geração precisa de SQL.

O alinhamento adequado entre essas capacidades tecnológicas e as necessidades dos usuários lusófonos é fundamental para garantir a efetividade do sistema na recuperação de dados e, consequentemente, no suporte à tomada de decisões organizacionais. Além disso, a implementação de uma solução de *Text-to-SQL* em português pode ter impactos significativos no suporte à tomada de decisão organizacional. Ao permitir que usuários não técnicos, como fiscais e gestores em órgãos públicos, acessem e analisem dados diretamente em sua língua nativa, a solução facilita decisões baseadas em dados integrados, reduz erros humanos na extração de informações e melhora a escalabilidade do acesso a dados críticos. Por exemplo, em um contexto de vigilância sanitária, essa capacidade pode acelerar

a identificação de padrões, como propriedades com baixa cobertura vacinal, otimizando ações preventivas e estratégicas.

Este estudo se propõe a investigar sistematicamente o desempenho de diferentes LLMs na tarefa de *Text-to-SQL* em português, utilizando como referência o Spider [32], um dos conjuntos de dados mais relevantes para avaliação dessa tecnologia. Por meio da análise comparativa entre o desempenho em português e inglês, busca-se compreender não apenas a viabilidade da aplicação desses modelos em contextos lusófonos, mas também identificar possíveis lacunas e oportunidades de aprimoramento.

A pesquisa contempla a avaliação de sete LLMs de diferentes capacidades e especializações, incluindo modelos generalistas e especializados em código. Essa seleção permite uma análise abrangente do estado atual da tecnologia e suas limitações, considerando aspectos como exatidão na geração de consultas, robustez na interpretação de instruções em português e consistência nas respostas. Além disso, como uma contribuição importante deste trabalho, foi realizada uma tradução minuciosa das 2.147 instâncias da partição *test* do Spider. Esse processo incluiu a revisão cuidadosa das traduções para garantir a fidelidade e aplicabilidade do *dataset* em contextos de língua portuguesa, o que é detalhado posteriormente neste artigo. O *dataset* traduzido encontra-se disponível publicamente[1].

Os resultados deste trabalho têm implicações significativas tanto para a pesquisa acadêmica quanto para aplicações práticas. Do ponto de vista acadêmico, contribui para a compreensão dos desafios específicos da transferência entre idiomas em tarefas de *Text-to-SQL* e estabelece métricas de referência para futuros desenvolvimentos. Na perspectiva prática, oferece *insights* para organizações que buscam implementar soluções de acesso a dados em português, auxiliando na seleção de modelos e identificação de possíveis limitações.

Além disso, este estudo se insere em um contexto mais amplo de democratização do acesso a dados e conhecimento, alinhando-se com iniciativas globais de redução de barreiras linguísticas no acesso à tecnologia. A disponibilidade de soluções *Text-to-SQL* eficientes em português tem o potencial de acelerar a adoção de práticas baseadas em dados em organizações lusófonas, contribuindo para o desenvolvimento tecnológico e econômico desses países.

Este estudo foi motivado pela necessidade da escolha de um LLM para ser usado para o desenvolvimento de um *chatbot* para consultas em linguagem natural a banco de dados, no contexto de um projeto em desenvolvimento para o Instituto Mineiro de Agropecuária (IMA)[2]. O projeto visa desenvolver uma plataforma que integre informações, promovendo gestão territorial e rastreabilidade mais eficientes, de forma sustentável e inovadora. Para isso, as informações precisam ser mais facilmente acessadas pelos *stakeholders* envolvidos, como fiscais, produtores rurais, empresas privadas, dentre outros. Após 32 anos de existência, o IMA possui várias bases de dados que dão suporte às atividades de fiscalização, inspeção de produtos de origem animal, certificação de produtos agropecuários e educação sanitária, gestão territorial, rastreabilidade, identificação de origens e o controle e erradicação de doenças.

[1]https://huggingface.co/datasets/Boakpe/spider-test-portuguese
[2]https://www.ima.mg.gov.br/

Para obter informação útil, é necessário que mais de uma base de dados seja consultada. Por exemplo, uma consulta às bases de dados de cadastro de fêmeas, de vacinações, de cadastro de propriedades e de guia de transporte de animais entre propriedades, permite identificar quais são as propriedades que não realizaram a vacinação de fêmeas. Do ponto de vista de um fiscal do IMA, essa informação é importante para se realizar a prevenção ou evitar a disseminação de doenças como a brucelose, que pode ser transmitida entre animais e para o ser humano (através de leite, derivados ou pela carne contaminada). Ter uma ferramenta de consulta usando linguagem natural pode facilitar muito o trabalho de um fiscal e gerar um impacto social positivo.

Nas seções seguintes, são apresentadas a fundamentação teórica que embasa o estudo, os métodos utilizados nos experimentos, a análise dos resultados obtidos e a discussão sobre suas implicações para o campo de *Text-to-SQL* em português. Além disso, são identificadas as limitações do estudo e sugeridas direções para futuras pesquisas, com o objetivo de aprimorar continuamente essa tecnologia para usuários lusófonos.

2 FUNDAMENTAÇÃO TEÓRICA

A tarefa de *Text-to-SQL* busca traduzir questões em linguagem natural para consultas SQL, facilitando a interação de usuários não especializados com bancos de dados e otimizando o processamento de dados em aplicações de análise automática e questionamento de bases de dados [22, 28, 31]. Tradicionalmente, abordagens para *Text-to-SQL* incluem tanto métodos baseados em regras quanto técnicas de aprendizado supervisionado, com destaque recente para o uso de modelos neurais e de aprendizado profundo [2, 19, 25]. Esses métodos avançaram a tarefa ao combinar modelos preditivos com conhecimentos contextuais, permitindo traduções mais precisas e flexíveis [21].

As abordagens de *Text-to-SQL* podem ser classificadas em dois tipos principais: (i) métodos baseados em regras, que dependem de interpretações de entidades e do uso de estados intermediários para gerar a consulta SQL final, com exemplos como NaLIR [14] e Athena [23]; e (ii) métodos baseados em aprendizado profundo, que incluem arquiteturas *encoder-decoder* e técnicas de atenção para otimizar a correspondência entre perguntas e SQL, com sistemas como Seq2SQL [34], SQLNet [29] e RAT-SQL [28].

Nos últimos anos, os *Large Language Models* têm surgido como uma abordagem promissora para *Text-to-SQL*, especialmente com o advento de modelos como o GPT-4 [18]. Os LLMs têm se mostrado eficazes em entender e gerar respostas contextualizadas em uma ampla gama de tarefas de linguagem natural. Em *Text-to-SQL*, o foco principal passou a ser a engenharia de *prompt*, isto é, a construção de representações de entrada para maximizar a eficácia das consultas SQL geradas [21]. Diferentes técnicas de *prompt* têm sido exploradas, como o uso de tabelas em formato SQL, exemplos explícitos, chaves estrangeiras e instruções de tarefas para guiar os LLMs na resposta correta [2, 19].

A engenharia de *prompt* no contexto de *Text-to-SQL* pode ser realizada em dois cenários principais: *zero-shot* e *few-shot*. Em cenários *zero-shot*, onde nenhum exemplo é fornecido, o desafio reside em representar a consulta natural de maneira a incorporar informações relevantes, como o esquema do banco de dados [5, 22]. Por outro

lado, em cenários *few-shot*, onde um conjunto limitado de exemplos é incluído, a seleção e organização dos exemplos se tornam fundamentais para potencializar a capacidade de aprendizado em contexto dos LLMs [17]. Estudos recentes evidenciam que a seleção cuidadosa de exemplos pode aumentar a eficácia na geração de SQL, sobretudo ao representar as relações semânticas entre pergunta e consulta alvo [6].

Outro aspecto relevante é o uso de modelos de código aberto, como o LLaMA da Meta [27] e outras arquiteturas recentes, que têm expandido as possibilidades de pesquisa e aplicação de *Text-to-SQL* fora do ambiente proprietário da OpenAI. Modelos de código aberto não apenas alcançam desempenho comparável aos modelos proprietários em diversas tarefas [4, 30], como também oferecem flexibilidade e adaptabilidade para diferentes cenários de implementação. Apesar disso, alguns deles, principalmente os de menor quantidade de parâmetros, ainda podem enfrentar desafios em relação à coerência contextual e à robustez em algumas tarefas específicas, o que pode ser abordado com técnicas como *supervised fine-tuning* (SFT). Essa abordagem de ajuste fino supervisionado é eficaz para melhorar a adequação do modelo à tarefa de *Text-to-SQL* e mitigar problemas de vieses e alucinações [35]. No entanto, com o avanço constante de modelos abertos e a evolução das técnicas de treinamento, esses desafios têm se tornado cada vez mais gerenciáveis, ampliando significativamente as aplicações desses modelos no campo da geração de consultas SQL.

Além disso, o custo e a eficiência de *tokens* permanecem como desafios no uso de LLMs para *Text-to-SQL*, uma vez que chamadas frequentes às *APIs* são caras e possuem limitações de taxa. Estudos indicam que o comprimento do *prompt* afeta a eficácia da execução das consultas, sugerindo que há um ponto ideal para a engenharia de *prompt* em termos de eficiência de *tokens* [33]. Em resposta a esses desafios, soluções como o DAIL-SQL foram propostas, combinando a seleção eficiente de exemplos e a minimização de *tokens* para atingir um equilíbrio entre custo e eficácia, estabelecendo novos padrões de desempenho no Spider *leaderboard* [6].

O estado da arte em *Text-to-SQL* permanece fundamentado no uso de LLMs, porém com uma evolução significativa nas técnicas de otimização. Além da engenharia de *prompt* e *fine-tuning*, as abordagens atuais incorporam o conceito de *test-time compute*, que envolve estratégias sofisticadas de geração e seleção de respostas [7, 20]. Uma técnica representativa é a geração de múltiplas consultas SQL para uma mesma pergunta utilizando diferentes variações de *prompts*, seguida pela aplicação de um modelo classificador que seleciona a resposta mais adequada. Embora essas técnicas avançadas demonstrem resultados significativamente superiores às abordagens convencionais anteriormente citadas, sua implementação prática em organizações apresenta limitações importantes. O alto custo computacional e os expressivos custos operacionais associados tornam essas soluções frequentemente inviáveis para implementação em larga escala em contextos organizacionais.

Portanto, enfatiza-se a importância de três aspectos fundamentais para o avanço dos modelos de *Text-to-SQL no contexto prático*: a engenharia de *prompt*, o aprendizado contextual e o ajuste fino supervisionado. Essas abordagens, quando combinadas com estratégias para otimização de *tokens*, oferecem uma base sólida para o desenvolvimento de soluções práticas e eficazes, possibilitando aplicações mais robustas e acessíveis economicamente na tarefa de tradução de linguagem natural para SQL.

3 METODOLOGIA

Esta seção apresenta a metodologia empregada neste estudo, com o detalhamento dos principais aspectos do processo experimental. A primeira subseção descreve o *dataset* Spider utilizado e seu processo de tradução para o português (Seção 3.1). Em seguida, são apresentados os sete modelos de linguagem selecionados para avaliação, suas características e processo de quantização (Seção 3.2). A configuração dos experimentos é detalhada na terceira subseção, com ênfase nas estratégias de *prompt* e abordagem *zero-shot* adotadas (Seção 3.3). Por fim, são apresentadas as métricas de avaliação utilizadas - Exact Match (EM) e Execution Accuracy (EX) - e sua relevância para a análise comparativa do desempenho dos modelos em português e inglês (Seção 3.4).

3.1 *Dataset* e Processo de Tradução

Para a condução dos experimentos, foi utilizado o *dataset* Spider [32], um dos *benchmarks* mais relevantes para tarefas de *Text-to-SQL*. O Spider compreende 200 bancos de dados distintos, já preenchidos com dados, abrangendo 138 domínios variados, com uma média de aproximadamente 51 consultas SQL por banco de dados. Cada instância do *dataset* é composta por:

- Uma pergunta em linguagem natural
- A consulta SQL correspondente à pergunta
- O esquema do banco de dados relacionado
- O nível de dificuldade da consulta (fácil, médio, difícil ou extra difícil)

Embora o Spider venha sendo amplamente utilizado como referência para avaliação de modelos e abordagens na área, é importante contextualizar que sua relativa simplicidade nem sempre reflete adequadamente os desafios encontrados em aplicações reais. Em resposta a essa limitação, *datasets* mais complexos foram desenvolvidos posteriormente, como o BIRD [15]. O BIRD (Big Bench for Large-scale Database Grounded Text-to-SQL Evaluation) é um conjunto de dados para avaliação de *Text-to-SQL* em larga escala e entre múltiplos domínios. Ele contém mais de 12.751 pares únicos de perguntas e consultas SQL, abrangendo 95 grandes bancos de dados, com um tamanho total de 33,4 GB. O BIRD cobre mais de 37 domínios profissionais. Atualmente, o BIRD está disponível apenas em inglês e, devido ao seu tamanho expressivo e complexidade, os custos e esforços para traduzi-lo para o português seriam muito altos.

Considerando que o objetivo principal deste estudo é analisar comparativamente o desempenho de modelos em português e inglês, o Spider se mostrou adequado para a avaliação, uma vez que sua complexidade é suficiente para estabelecer as diferenças de performance entre os dois idiomas.

O Spider é estruturado em três partições:

- Treino (train): 7.000 instâncias (68,76%)
- Desenvolvimento (dev): 1.034 instâncias (10,16%)
- Teste (test): 2.147 instâncias (21,09%)

Como o *dataset* foi originalmente desenvolvido para *Text-to-SQL* em inglês, foi necessário realizar sua tradução para o português. As

partições de treino e desenvolvimento foram previamente traduzidas por [13]. A partição de teste, disponibilizada posteriormente pela equipe do Spider, ainda não possuía uma tradução. Para atender a essa necessidade, a tradução desta parte foi realizada neste trabalho utilizando a *API* do GPT-4o mini da OpenAI, seguida por um rigoroso processo de revisão e validação manual de 2.147 questões.

O processo de tradução foi conduzido com os seguintes critérios:

- Preservação máxima da estrutura sintática das questões originais
- Manutenção de termos técnicos sem equivalentes consolidados em português (e.g., *"powertrain"*)
- Conservação de valores em inglês quando presentes no banco de dados, mesmo havendo traduções possíveis para o português (e.g., *"United States"* foi mantido em vez de *"Estados Unidos"*)

Esses critérios foram estabelecidos para garantir que o modelo pudesse identificar corretamente as correspondências entre os termos nas consultas e os valores presentes no banco de dados, minimizando assim potenciais ambiguidades na geração das consultas SQL.

3.2 Modelos Utilizados

Para a realização deste estudo, foram selecionados sete LLMs. A seleção priorizou modelos que podem ser efetivamente implementados por instituições e empresas em ambientes produtivos, excluindo assim modelos como o Mistral 7B [12] que, apesar de sua popularidade na comunidade acadêmica, apresentam restrições de licenciamento para uso comercial.

3.2.1 Modelos de Base (~8B parâmetros). Primeiramente, foram selecionados três modelos generalistas *open-source* com aproximadamente 8 bilhões de parâmetros:

- Qwen 2.5 7B [26]
- Llama 3.1 8B [4]
- Granite 3 8B [11]

A escolha desses modelos foi motivada pela necessidade de avaliar o desempenho de arquiteturas que podem ser executadas em hardware de uso geral, como computadores pessoais com capacidade computacional moderada.

3.2.2 Modelos Especializados em Código.

- Qwen 2.5 Coder 7B [10]
- Qwen 2.5 Coder 32B

Esses modelos foram incluídos com o objetivo de investigar as possíveis diferenças de desempenho entre modelos generalistas e aqueles especificamente otimizados para tarefas de programação.

3.2.3 Modelos Generalistas de Maior Escala.

- Qwen 2.5 32B [26]
- GPT-4o Mini

Esses modelos foram selecionados com o objetivo de realizar comparações com arquiteturas de maior capacidade. O Qwen 2.5 32B foi escolhido por ser o maior modelo que pode ser executado em uma GPU RTX 3090 com 24 GB de VRAM, permitindo uma comparação direta com o GPT-4o Mini, o modelo proprietário de menor porte disponível pela OpenAI, que se adequava ao orçamento deste estudo.

3.2.4 Quantização. Para os modelos de aproximadamente 8B parâmetros, foi aplicada a quantização Q8_0 do `llama.cpp`. Esta decisão foi fundamentada em estudos anteriores [9, 16] que demonstram que:

- O desempenho dos modelos quantizados é praticamente equivalente aos modelos com precisão completa
- A velocidade de inferência é significativamente superior nos modelos quantizados

Para o Qwen 2.5 32B, foi aplicada a quantização Q4_K_M, visando otimizar o uso de memória e permitir que o modelo se ajuste aos 24 GB de VRAM da RTX 3090, ao mesmo tempo em que mantém um nível satisfatório de qualidade nas predições.

Essa seleção de modelos permite avaliar compreensivamente o desempenho de LLMs na tarefa de *Text-to-SQL* em português, considerando diferentes escalas de parâmetros, especializações e requisitos computacionais. A inclusão de modelos de diferentes capacidades também possibilita uma análise do *trade-off* entre tamanho do modelo e qualidade das predições.

3.3 Configuração dos Experimentos

Diversos fatores influenciam o desempenho de LLMs em tarefas de *Text-to-SQL*, sendo a disponibilidade e o formato das informações fornecidas ao modelo os mais críticos [6]. Embora a literatura apresente diferentes estratégias de construção de *prompts* [33], este estudo prioriza a consistência metodológica para uma comparação precisa do desempenho entre modelos em português e inglês. Portanto, em vez de buscar o *prompt* ideal para o português, foi optado por um *prompt* único, adaptado do trabalho de [1], para todos os modelos avaliados. Um exemplo desse *prompt* é ilustrado na Figura 1. Essa abordagem, embora possa não maximizar o desempenho individual em cada idioma, garante maior controle experimental e permite uma análise comparativa mais robusta.

Para cada consulta, o *prompt* inclui, além da pergunta em linguagem natural, o esquema completo do banco de dados associado. Isso significa que, para cada uma das 50 instâncias (consultas) associadas a um determinado banco de dados, o mesmo esquema (definição das tabelas, colunas e seus tipos de dados) é repetido no *prompt*. Essa abordagem garante que o modelo tenha acesso consistente a todas as informações estruturais necessárias para gerar a consulta SQL correta.

Adicionalmente, outra estratégia comum na literatura é a abordagem *few-shot* [6], em que o modelo recebe exemplos de questões e respostas para o problema a ser resolvido. No entanto, neste estudo foi adotada uma abordagem *zero-shot*, ou seja, sem fornecer exemplos ao modelo. Essa escolha foi motivada por dois fatores principais: primeiramente, o objetivo do estudo é avaliar as diferenças de desempenho entre os modelos em português e em inglês, e não necessariamente obter o melhor desempenho possível em português. Em segundo lugar, a inclusão de exemplos no *prompt* aumentaria consideravelmente o número de *tokens* dentro da janela de contexto [33], o que, por sua vez, impactaria negativamente o tempo de execução dos testes.

Além disso, para garantir uma resposta consistente e objetiva dos modelos, a temperatura foi configurada para 0 (zero) durante

os experimentos. Essa configuração foi escolhida para minimizar a aleatoriedade nas respostas e garantir que os modelos fornecessem respostas mais determinísticas, o que é especialmente importante em tarefas de *Text-to-SQL*, onde a eficácia da consulta SQL gerada é crucial.

```
CREATE TABLE club (
    Club_ID int PRIMARY KEY,
    Name text,
    Manager text,
    Captain text,
    Manufacturer text,
    Sponsor text
);

CREATE TABLE player (
    Player_ID real PRIMARY KEY,
    Name text,
    Country text,
    Earnings real,
    Events_number int,
    Wins_count int,
    Club_ID int,
    FOREIGN KEY (Club_ID) REFERENCES club(Club_ID)
);

-- Usando SQLite válido, responda à seguinte
    pergunta para as tabelas fornecidas acima.
    Responda apenas com uma query SQL sem nenhuma
    explicação.

Pergunta: Qual é o país do jogador com os maiores
    ganhos entre os jogadores que têm mais de 2
    vitórias?
Somente uma query SQL:
```

Figura 1: Exemplo de *prompt* utilizado para avaliação dos modelos em português

3.4 Métricas de Avaliação

A avaliação quantitativa dos modelos foi realizada utilizando as duas métricas principais estabelecidas pelo Spider:

- **Exact Match (EM)**: Avalia a correspondência sintática entre a consulta SQL gerada pelo modelo e a consulta de referência. Esta métrica incorpora um sistema de normalização que reconhece variações sintáticas equivalentes na linguagem SQL. Por exemplo, em casos onde a tabela é implícita, as consultas 'SELECT name' e 'SELECT name FROM player' são consideradas correspondências exatas. Embora o EM seja uma métrica mais restritiva, ela fornece *insights* valiosos sobre a capacidade do modelo em reproduzir a estrutura sintática esperada e pode indicar possíveis efeitos de memorização durante o processo de *fine-tuning*.
- **Execution Accuracy (EX)**: Mensura a equivalência semântica entre as consultas através da comparação dos resultados de sua execução. Essa métrica é considerada mais robusta por reconhecer que diferentes estruturas sintáticas de SQL podem produzir conjuntos de resultados idênticos. Por exemplo, subconsultas podem ser reescritas como JOINs mantendo a mesma semântica.

Entre as duas métricas apresentadas, a *Execution Accuracy* emerge como o indicador mais relevante para avaliação de desempenho no contexto do *benchmark* Spider. Essa preferência se justifica pela natureza da linguagem SQL, que permite múltiplas formas sintaticamente distintas de expressar a mesma consulta semântica. Enquanto a EM oferece uma perspectiva complementar, particularmente útil para análises de *overfitting* e padrões de aprendizagem do modelo, a EX fornece uma avaliação mais precisa da capacidade do modelo em gerar consultas funcionalmente corretas.

Essa abordagem dual de avaliação permite uma análise mais abrangente do desempenho dos modelos, considerando tanto aspectos sintáticos quanto semânticos da geração de SQL. Como será discutido posteriormente neste artigo, a análise conjunta dessas métricas revela padrões importantes sobre o comportamento dos modelos e sua capacidade de generalização.

4 RESULTADOS

4.1 Desempenho Geral

As Tabelas 1 e 2 apresentam uma análise detalhada do desempenho dos modelos selecionados nas métricas de *Exact Match* (EM) e *Execution Accuracy* (EX) para consultas em português e inglês, nas partições *dev* e *test*, respectivamente.

Embora a partição *test* seja a utilizada para a classificação no *leaderboard* do *Spider*, optou-se também por incluir os resultados na partição *dev*. Essa escolha se justifica pelo fato de a partição *dev* ser historicamente a primeira disponibilizada, sendo amplamente utilizada em trabalhos mais antigos. Somente em um momento posterior foi liberada a partição *test* para avaliação oficial.

Modelo	Português		Inglês	
	EM (%)	EX (%)	EM (%)	EX (%)
Qwen 2.5 7B	38,2	62,8	40,5	72,5
Llama 3.1 8B	27,2	62,3	34,8	71,1
Granite 3 8B	28,8	54,7	27,8	62,8
Qwen 2.5 Coder 7B	**56,2**	72,0	**66,7**	80,8
Qwen 2.5 Coder 32B	53,9	**75,0**	61,6	**83,8**
Qwen 2.5 32B	41,7	69,1	33,7	78,1
GPT-4o Mini	26,9	69,6	26,7	76,4

Tabela 1: Desempenho dos modelos nas métricas EM e EX para consultas em português e inglês na partição *dev*.

Os resultados demonstram um desempenho superior dos modelos Qwen 2.5 Coder 7B e 32B , Qwen 2.5 32B e GPT-4o Mini nas métricas EM e EX, em ambas as línguas avaliadas. Esse comportamento era esperado, dado que modelos de linguagem desenvolvidos com especialização para tarefas de codificação, bem como aqueles de maior capacidade (maior número de parâmetros), tendem a alcançar melhores resultados no *benchmark Spider* [32].

Além disso, observa-se uma disparidade significativa nos valores de EM entre os modelos da série Qwen e os demais modelos. Esse desempenho pode ser atribuído, em parte, ao treinamento específico

Modelo	Português		Inglês	
	EM (%)	EX (%)	EM (%)	EX (%)
Qwen 2.5 7B	42,4	72,0	43,7	74,8
Llama 3.1 8B	31,3	68,0	35,6	70,4
Granite 3 8B	29,7	52,9	31,3	69,2
Qwen 2.5 Coder 7B	**57,0**	77,7	**62,1**	79,5
Qwen 2.5 Coder 32B	56,9	**82,0**	**62,1**	**83.7**
Qwen 2.5 32B	44,8	77,0	36,4	78,1
GPT-4o Mini	29,3	76,4	30,2	77,1

Tabela 2: Desempenho dos modelos nas métricas EM e EX para consultas em português e inglês na partição *test*.

desses modelos, que incluiu dados da partição de treino do conjunto *Spider*. Essa hipótese é sustentada por dois aspectos principais: primeiro, o valor elevado de EM, que é comumente associado a modelos que passaram por *fine-tuning* ou treinamento com dados similares ao do conjunto de avaliação; segundo, o fato da equipe de desenvolvimento dos modelos Qwen ter referenciado o *Spider* no relatório técnico do modelo [10].

4.2 Diferença entre Português e Inglês

Os resultados obtidos indicam que, de maneira geral, modelos de maior desempenho em *Text-to-SQL* apresentam uma menor diferença de performance entre as línguas portuguesa e inglesa. Em termos empíricos, isso sugere que, conforme se aumenta a capacidade dos modelos, a discrepância de desempenho entre os dois idiomas tende a diminuir, aproximando-se dos resultados observados em inglês. No entanto, para que essa observação possa ser formalizada como uma conclusão robusta, seriam necessários *benchmarks* adicionais que validassem o desempenho dos modelos especificamente em português. Embora já existam alguns testes nesse sentido [24], eles ainda não estão amplamente disponíveis.

Observa-se uma disparidade significativa no desempenho entre português e inglês na partição *dev*. Como essa partição foi traduzida para o português por outros autores, não é possível garantir total consistência nos critérios de tradução empregados, o que pode ter influenciado a interpretação das perguntas em português pelos modelos de linguagem. Devido a essa incerteza, a análise foi focada exclusivamente na partição *test*, que foi revisada e traduzida de acordo com critérios consistentes neste estudo.

Para explorar a hipótese de que modelos mais robustos tendem a reduzir a diferença de desempenho entre inglês e português de forma mais substancial, foi realizada uma análise com diferentes tamanhos de modelo de uma mesma família, investigando a relação entre capacidade do modelo e diferença de desempenho entre os idiomas. A abordagem selecionada foi avaliar seis modelos da família Qwen 2.5 [26], com diferentes quantidades de parâmetros: Qwen 2.5-0.5B, Qwen 2.5-1.5B, Qwen 2.5-3B, Qwen 2.5-7B, Qwen 2.5-14B e Qwen 2.5-32B. Essa escolha foi estratégica, pois permite observar o impacto do aumento de capacidade (em número de parâmetros) mantendo constantes tanto a arquitetura dos modelos quanto o conjunto de dados de treinamento. Como todos os modelos foram treinados com a mesma base de dados, foi assumido que a exposição a dados em português foi equivalente para todos, isolando a capacidade do modelo como a principal variável em estudo.

Os experimentos foram conduzidos na partição de teste do *dataset* Spider, composta por 2.147 perguntas, para avaliar o desempenho das variantes de Qwen em ambas as línguas. Em seguida, calculou-se a diferença de desempenho em termos da métrica *Execution Accuracy* (EX) entre as respostas em português e inglês. Observou-se, a partir dos resultados, uma tendência de redução da diferença de desempenho com o aumento do número de parâmetros do modelo. Essa observação foi corroborada pelo gráfico na Figura 2, que indica uma correlação negativa entre a quantidade de parâmetros e a discrepância de performance entre as línguas. Assim, infere-se que modelos de maior capacidade possuem uma habilidade aprimorada de adaptação às variações linguísticas, o que é particularmente relevante no contexto de transferência entre idiomas, como entre inglês e português.

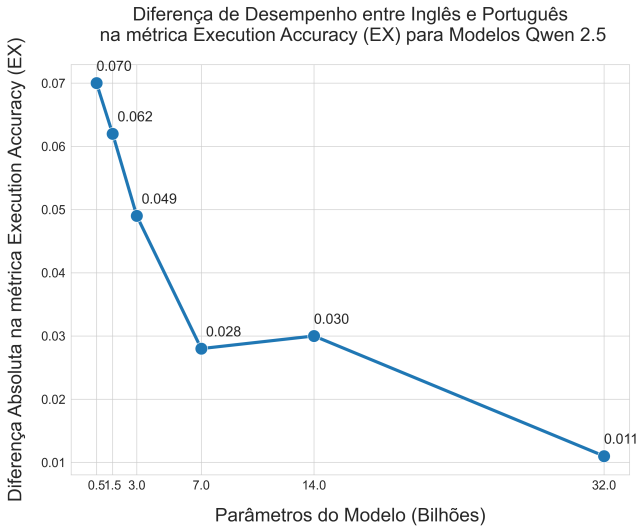

Figura 2: Diferença de desempenho entre Inglês e Português na métrica EX para modelos da família Qwen 2.5

A partir desses resultados, é razoável sugerir que a alta capacidade dos modelos permite a internalização de representações linguísticas mais universais, o que favorece um desempenho consistente em múltiplos idiomas. Embora essa tendência de redução na diferença entre idiomas tenha sido observada de maneira significativa na família Qwen 2.5, mais estudos são necessários para verificar se essa correlação se mantém para outras famílias de modelos e diferentes arquiteturas. A ausência de *benchmarks* especializados em português para *Text-to-SQL* limita a generalização dos resultados, pois não é possível validar esses achados em contextos mais desafiadores que envolvam linguagem técnica ou variações semânticas e sintáticas mais complexas.

Portanto, os resultados sugerem que, na tarefa de *Text-to-SQL*, a capacidade de um modelo parece correlacionada à sua habilidade de minimizar diferenças de desempenho entre idiomas.

4.3 Diferentes Respostas

Durante a execução dos experimentos, observou-se que certos modelos de LLM não geravam exclusivamente a consulta SQL solicitada, mas sim uma resposta completa que incluía uma explicação

adicional sobre a consulta. Esse comportamento apresentou variação significativa entre os modelos testados, sendo mais frequente nos modelos generalistas de menor número de parâmetros. Os modelos da família Qwen 2.5 Coder, especificamente otimizados para programação, foram os únicos que consistentemente retornaram apenas a consulta SQL sem nenhuma explicação ou comentário adicional.

Esse fenômeno pode ser explicado pelo treinamento e pelo ajuste fino dos LLMs generalistas, que frequentemente recebem instruções para responder perguntas de maneira detalhada. Nos experimentos realizados, verificou-se que esse comportamento indesejado se manifestava com maior frequência em questões de maior complexidade, bem como em *prompts* contendo um elevado número de *tokens*, como no caso de bases de dados com um grande número de tabelas. Em consultas SQL mais simples, os modelos generalistas demonstraram maior propensão a seguir as instruções de retorno direto da consulta, ao passo que, em consultas de complexidade elevada, o número de respostas que incluíam explicações adicionais aumentou.

Dado que para aplicações *Text-to-SQL* em ambientes produtivos é desejável que o modelo forneça apenas a consulta SQL, essa característica dos modelos generalistas pode representar um obstáculo operacional. A necessidade de isolar a consulta SQL de uma resposta mais extensa implica um processo adicional de pós-processamento que pode adicionar uma camada de complexidade e suscitar possíveis fontes de erro. Por exemplo, respostas que não contêm a consulta SQL em um bloco de código *markdown* ou envolta em *tags* podem dificultar a extração precisa e automatizada do comando SQL. Esse problema é especialmente relevante em tarefas onde a consulta gerada deve ser diretamente utilizada para consultas no banco de dados, pois o processo de filtragem de explicações ou comentários torna-se crucial para evitar respostas incorretas ou parciais.

Para mitigar este comportamento, foram testadas múltiplas variações de *prompts* com o objetivo de maximizar a adesão dos modelos à instrução de retornar apenas a consulta SQL. Os resultados empíricos indicam que alguns modelos são sensíveis à formulação das consultas, o que permite uma redução parcial na ocorrência de respostas extensas, embora essa abordagem nem sempre seja eficaz. A persistência desse problema sugere que para certas implementações de *Text-to-SQL*, especialmente onde o retorno direto da consulta SQL é imprescindível, modelos especializados para tarefas de código apresentam uma vantagem operacional. Além disso, a adoção de um processo pós-executivo de verificação pode ser necessário para assegurar que a resposta gerada consiste unicamente na consulta SQL.

4.4 Desafios do Banco de Dados Real e Estratégias de Adaptação

A aplicação de técnicas de *Text-to-SQL* em ambientes de produção apresenta desafios que vão além daqueles representados em *benchmarks* acadêmicos, como o Spider. A utilização direta do banco de dados real do IMA revelou obstáculos significativos, levando à adoção de uma abordagem em etapas, iniciando com o *benchmark* Spider.

No contexto do banco de dados analisado, dois desafios principais foram identificados:

(1) **Complexidade Estrutural:** O banco de dados examinado contém aproximadamente 600 tabelas, uma escala significativamente maior do que a maioria dos *datasets* utilizados em pesquisas de *Text-to-SQL*. Essa complexidade amplia o espaço de busca dos LLMs, podendo impactar negativamente tanto o tempo de resposta quanto a precisão das consultas geradas. A grande quantidade de tabelas também eleva a probabilidade de geração de consultas que, embora sintaticamente corretas, não correspondam à semântica desejada.
(2) **Problemas de Estruturação:** A organização dos dados no banco de dados não segue rigorosamente as melhores práticas de modelagem, dificultando a interpretação semântica das tabelas e seus relacionamentos pelos LLMs. Esse fator pode resultar em erros na geração de consultas SQL, especialmente em perguntas mais complexas que envolvem múltiplas tabelas. Problemas estruturais identificados incluem a ausência de chaves estrangeiras bem definidas, nomes de colunas pouco descritivos e a falta de documentação clara do esquema do banco de dados.

Para mitigar esses desafios e viabilizar a aplicação da tecnologia *Text-to-SQL* em um contexto real, foram propostas as seguintes estratégias de adaptação:

(1) **Criação de *Views*:** A implementação de *views* no banco de dados possibilita a criação de uma camada de abstração sobre a estrutura original. Essas *views* podem ser projetadas para organizar os dados de forma mais coerente, facilitando a interpretação pelos LLMs. Por exemplo, *views* podem ser utilizadas para combinar informações de múltiplas tabelas relacionadas, reduzindo a necessidade de *joins* complexos. Além disso, essa abordagem permite renomear colunas com nomes mais descritivos e ocultar informações irrelevantes para a aplicação *Text-to-SQL*.
(2) **Seleção Criteriosa de Tabelas:** Em vez de expor a totalidade das tabelas aos LLMs, um processo de análise é realizado em conjunto com especialistas no domínio para identificar as tabelas mais relevantes para as necessidades dos usuários. Essa seleção reduz o escopo do problema, melhorando a eficiência e a precisão dos modelos, além de possibilitar a construção de *prompts* mais concisos e direcionados. O processo ocorre de maneira iterativa, iniciando com um subconjunto reduzido de tabelas e expandindo conforme necessário.
(3) **Engenharia de *Prompt* Específica:** Além das adaptações no banco de dados, são exploradas técnicas de engenharia de *prompt* específicas para o contexto em questão. Entre essas técnicas, incluem-se a inclusão de exemplos de perguntas e consultas SQL representativas do domínio, a incorporação de metadados sobre tabelas e colunas (como descrições e tipos de dados) e a definição de restrições ou preferências na formulação das consultas.

O uso do *benchmark* Spider proporciona uma avaliação inicial da tecnologia em um ambiente controlado, permitindo identificar limitações e ajustar estratégias antes de sua aplicação ao banco de dados real. Esse processo facilita a transição para um contexto mais complexo, no qual questões relacionadas à governança de dados,

escalabilidade e adaptação da modelagem tornam-se ainda mais relevantes.

4.5 Impactos Organizacionais da Solução Proposta

A partir dos resultados obtidos, é possível prever impactos positivos para organizações lusófonas, como o IMA. Sob a perspectiva das necessidades dos usuários lusófonos, a capacidade de formular consultas em português democratiza o acesso a informações críticas, especialmente para profissionais sem expertise em SQL ou programação, como fiscais e técnicos. Esse avanço promove a *inclusão operacional*, ampliando o número de usuários capazes de extrair *insights* diretamente do banco de dados, sem depender de relatórios pré-definidos ou intermediários técnicos. Além disso, assegura maior *precisão contextual* nas consultas, reduzindo ambiguidades linguísticas e alinhando as perguntas aos esquemas do banco de dados, o que é essencial em contextos onde a fidelidade dos dados impacta resultados práticos, como o controle de epidemias. Por fim, a *eficiência* operacional é significativamente aumentada, pois a redução da dependência de especialistas acelera processos, como fiscalizações sanitárias em campo.

No que tange ao suporte à tomada de decisão organizacional, a solução proposta mitiga limitações de sistemas legados, que frequentemente exigem soluções improvisadas e consomem tempo considerável para a extração de dados. A implementação de um *chatbot* baseado em *Text-to-SQL* permite a recuperação direta de informações, seja para análises *ad hoc*, geração de visualizações ou exportação de dados. Isso resulta em *decisões baseadas em dados integrados*, possibilitando análises transversais que revelam padrões críticos — por exemplo, a identificação de propriedades com baixa cobertura vacinal para ações de prevenção. A *redução de erros humanos* na manipulação de dados aumenta a confiabilidade das informações usadas por gestores, enquanto a *escalabilidade* da solução permite que *stakeholders* diversificados, como fiscais, produtores e administradores, acessem dados de forma autônoma, otimizando o fluxo de trabalho e a capacidade de resposta organizacional.

4.6 Limites do Estudo

Este estudo, embora abrangente em seu escopo de avaliação de LLMs para *Text-to-SQL* em português e inglês, apresenta algumas limitações que devem ser reconhecidas. Em primeiro lugar, a análise incluiu um número limitado de modelos, principalmente devido a restrições orçamentárias e computacionais. Modelos de maior capacidade, como GPT-4o, Claude 3.5 Sonnet e Gemini 1.5 Pro, que representam o estado da arte no campo de LLMs, não foram avaliados, o que limita a generalização dos resultados para outras arquiteturas de LLMs que poderiam potencialmente oferecer desempenho superior. A inclusão desses modelos avançados poderia proporcionar uma visão mais detalhada sobre o desempenho de LLMs de maior capacidade e suas capacidades de generalização em *Text-to-SQL*.

Além disso, a escolha do *dataset* Spider, embora representativa, apresenta limitações de complexidade. Estudos recentes sugerem que o Spider, por ser menos complexo que *benchmarks* mais recentes, como o BIRD, pode não refletir totalmente os desafios enfrentados em cenários de *Text-to-SQL* mais avançados. Dada a natureza relativamente padronizada e bem documentada das consultas no Spider, a tradução para o português, embora útil, pode não representar o desafio total de linguagens SQL aplicadas em contextos industriais mais complexos. A utilização do BIRD, um *dataset* mais elaborado que contempla consultas de maior complexidade e estrutura semântica, poderia oferecer uma visão mais precisa das dificuldades que modelos LLM enfrentariam ao interpretar instruções SQL em diferentes idiomas. Entretanto, isso requer a tradução do BIRD para o idioma português.

Por fim, uma limitação importante do estudo foi a ausência de um LLM especificamente treinado ou ajustado com o *dataset* Spider traduzido para o português. Embora este trabalho tenha adotado uma abordagem comparativa entre modelos em inglês e português, o uso de um modelo com *fine-tuning* para *Text-to-SQL* em português poderia evidenciar padrões de adaptação e aprimoramento na tarefa específica. Modelos que passam por esse processo de ajuste fino geralmente são capazes de incorporar nuances linguísticas e adaptar melhor as estruturas de consulta SQL ao idioma alvo. A inclusão de um LLM treinado especificamente para *Text-to-SQL* em português ampliaria o escopo da análise e permitiria uma comparação mais precisa entre os desempenhos nos dois idiomas.

Essas limitações sugerem oportunidades para futuras pesquisas, incluindo a avaliação de uma gama mais ampla de modelos, a aplicação de *benchmarks* mais complexos e a exploração de modelos ajustados especificamente para o idioma português. Esses avanços poderiam enriquecer substancialmente a análise de desempenho e o impacto do idioma em tarefas de *Text-to-SQL*, fornecendo uma visão mais robusta e abrangente sobre as capacidades e limitações dos LLMs nesse domínio.

5 CONCLUSÃO

Este estudo examinou o desempenho de diferentes *Large Language Models* na tarefa de *Text-to-SQL* em português, fornecendo *insights* valiosos sobre a viabilidade e eficácia dessa tecnologia para usuários lusófonos. Os resultados demonstraram que modelos como Qwen 2.5 Coder 7B e 32B, Qwen 2.5 32B e GPT-4o Mini apresentaram desempenho promissor na tradução de consultas em português para SQL, embora com algumas diferenças em relação ao seu desempenho em inglês.

Uma descoberta significativa foi a correlação inversa entre a capacidade dos modelos e a disparidade de desempenho entre os idiomas. Conforme observado na análise da família Qwen 2.5, modelos com maior número de parâmetros tenderam a apresentar menor diferença de performance entre português e inglês, sugerindo que o aumento da capacidade dos modelos contribui para uma maior robustez na transferência entre idiomas.

Os resultados também revelaram características importantes sobre o comportamento dos diferentes tipos de modelos. Modelos especializados em código, como os da família Qwen 2.5 Coder, demonstraram maior consistência na geração exclusiva de consultas SQL, enquanto modelos generalistas frequentemente incluíam explicações adicionais em suas respostas, o que pode representar um desafio para implementações em ambientes produtivos.

No entanto, como já mencionado, este estudo apresenta limitações importantes que devem ser consideradas. A ausência de modelos de maior capacidade como GPT-4o e Claude 3.5 Sonnet

na análise, bem como o uso do *dataset* Spider em vez de *benchmarks* mais complexos como o BIRD, podem limitar a generalização dos resultados. Além disso, a falta de um modelo especificamente ajustado para *Text-to-SQL* em português representa uma lacuna significativa na compreensão do potencial máximo dessa tecnologia para o idioma.

Sob a perspectiva teórica da Task-Technology Fit (TTF), este estudo oferece uma valiosa avaliação sobre a adequação dos LLMs para apoiar tarefas de *Text-to-SQL* em português. A TTF enfatiza a importância do alinhamento entre as capacidades tecnológicas e as necessidades da tarefa, e os resultados deste estudo indicam que modelos com maior capacidade e especialização em código se alinham melhor com as necessidades específicas de consultas SQL precisas em português. Isso sugere que, ao escolher a tecnologia adequada para suas necessidades, as organizações lusófonas podem alcançar uma implementação mais eficaz e eficiente de soluções de dados que suportam suas operações e decisões estratégicas.

A construção de um *chatbot* utilizando LMM para *Text-to-SQL* em português gera impacto organizacional para o IMA. Consultas não previstas em relatórios do sistema poderão ser feitas de forma "ad hoc", gerando mais agilidade para os usuários. Um fiscal em campo, por exemplo, pode abrir o celular e fazer perguntas específicas sobre a propriedade que está sendo fiscalizada, facilitando a inspeção de produtos de origem animal ou o controle da vacinação, dentre várias outras aplicações. O projeto em andamento entra, agora, na fase de avaliação dos LLMs nos casos reais das bases de dados do órgão público.

Este estudo não apenas avança o conhecimento técnico sobre *Text-to-SQL* em português, mas também destaca seu potencial impacto organizacional em contextos lusófonos. Ao facilitar o acesso a dados para usuários não técnicos, como os fiscais do IMA, a solução pode transformar processos decisórios, promovendo uma cultura de decisões baseadas em dados e aumentando a agilidade em cenários críticos, como fiscalizações sanitárias. Por exemplo, a implementação de um *chatbot* de *Text-to-SQL* permitiria que fiscais em campo acessassem informações em tempo real via dispositivos móveis, otimizando inspeções e acelerando respostas a emergências sanitárias, com benefícios diretos para a saúde pública.

Para pesquisas futuras, sugere-se as seguintes direções: (1) o desenvolvimento de um *dataset* nativo em português para *Text-to-SQL*, incluindo tanto as perguntas quanto as estruturas de banco de dados; (2) a avaliação de modelos de maior capacidade e sua performance em português; (3) a investigação do impacto do *fine-tuning* específico para português na tarefa de *Text-to-SQL*; e (4) a exploração de técnicas de *prompting* mais eficazes para reduzir a inclusão de conteúdo explicativo não solicitado nas respostas dos modelos generalistas.

Agradecimento

Este trabalho foi financiado pelo Instituto Mineiro de Agropecuária (IMA), Convênio nº 24/2024, celebrado entre a Universidade Federal de Lavras e o Instituto Mineiro de Agropecuária.

Referências

[1] Shuaichen Chang and Eric Fosler-Lussier. 2023. How to Prompt LLMs for Text-to-SQL: A Study in Zero-shot, Single-domain, and Cross-domain Settings. arXiv:2305.11853 [cs.CL] https://arxiv.org/abs/2305.11853

[2] Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Binding Language Models in Symbolic Languages. arXiv:2210.02875 [cs.CL] https://arxiv.org/abs/2210.02875

[3] Longxu Dou, Yan Gao, Mingyang Pan, Dingzirui Wang, Wanxiang Che, Dechen Zhan, and Jian-Guang Lou. 2023. MultiSpider: towards benchmarking multilingual text-to-SQL semantic parsing. In *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, Article 1430, 9 pages. https://doi.org/10.1609/aaai.v37i11.26499

[4] Abhimanyu Dubey and et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[5] Yujian Gan, Xinyun Chen, Jinxia Xie, Matthew Purver, John R. Woodward, John Drake, and Qiaofu Zhang. 2021. Natural SQL: Making SQL Easier to Infer from Natural Language Specifications. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Punta Cana, Dominican Republic, 2030–2042. https://doi.org/10.18653/v1/2021.findings-emnlp.174

[6] Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2024. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. *Proc. VLDB Endow.* 17, 5 (May 2024), 1132–1145. https://doi.org/10.14778/3641204.3641221

[7] Yingqi Gao, Yifu Liu, Xiaoxia Li, Xiaorong Shi, Yin Zhu, Yiming Wang, Shiqi Li, Wei Li, Yuntao Hong, Zhiling Luo, Jinyang Gao, Liyu Mou, and Yu Li. 2025. A Preview of XiYan-SQL: A Multi-Generator Ensemble Framework for Text-to-SQL. arXiv:2411.08599 [cs.AI] https://arxiv.org/abs/2411.08599

[8] Dale L. Goodhue and Ronald L. Thompson. 1995. Task-technology fit and individual performance. *MIS Q.* 19, 2 (June 1995), 213–236. https://doi.org/10.2307/249689

[9] Wei Huang, Xingyu Zheng, Xudong Ma, Haotong Qin, Chengtao Lv, Hong Chen, Jie Luo, Xiaojuan Qi, Xianglong Liu, and Michele Magno. 2024. An Empirical Study of LLaMA3 Quantization: From LLMs to MLLMs. arXiv:2404.14047 [cs.LG] https://arxiv.org/abs/2404.14047

[10] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, An Yang, Rui Men, Fei Huang, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. 2024. Qwen2.5-Coder Technical Report. arXiv:2409.12186 [cs.CL] https://arxiv.org/abs/2409.12186

[11] IBM. 2024. Granite 3.0 Language Models. https://github.com/ibm-granite/granite-3.0-language-models/blob/main/paper.pdf

[12] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[13] Marcelo Archanjo José and Fabio Gagliardi Cozman. 2021. *mRAT-SQL+GAP: A Portuguese Text-to-SQL Transformer*. Springer International Publishing, 511–525. https://doi.org/10.1007/978-3-030-91699-2_35

[14] Fei Li and Hosagrahar V Jagadish. 2014. NaLIR: an interactive natural language interface for querying relational databases. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data* (Snowbird, Utah, USA) *(SIGMOD '14)*. Association for Computing Machinery, New York, NY, USA, 709–712. https://doi.org/10.1145/2588555.2594519

[15] Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C.C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. 2024. 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* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 1835, 28 pages. https://dl.acm.org/doi/abs/10.5555/3666122.3667957

[16] Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. 2024. Evaluating Quantized Large Language Models. arXiv:2402.18158 [cs.CL] https://arxiv.org/abs/2402.18158

[17] Linyong Nan, Yilun Zhao, Weijin Zou, Narutatsu Ri, Jaesung Tae, Ellen Zhang, Arman Cohan, and Dragomir Radev. 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*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 14935–14956. https://doi.org/10.18653/v1/2023.findings-emnlp.996

[18] OpenAI and et al. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[19] Gabriel Poesia, Oleksandr Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. 2022. Synchromesh: Reliable code generation from pre-trained language models. arXiv:2201.11227 [cs.LG] https:

//arxiv.org/abs/2201.11227

[20] Mohammadreza Pourreza, Hailong Li, Ruoxi Sun, Yeounoh Chung, Shayan Talaei, Gaurav Tarlok Kakkar, Yu Gan, Amin Saberi, Fatma Ozcan, and Sercan O. Arik. 2024. CHASE-SQL: Multi-Path Reasoning and Preference Optimized Candidate Selection in Text-to-SQL. arXiv:2410.01943 [cs.LG] https://arxiv.org/abs/2410.01943

[21] Mohammadreza Pourreza and Davood Rafiei. 2024. DIN-SQL: decomposed in-context learning of text-to-SQL with self-correction. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 1577, 10 pages. https://dl.acm.org/doi/10.5555/3666122.3667699

[22] Ohad Rubin and Jonathan Berant. 2021. SmBoP: Semi-autoregressive Bottom-up Semantic Parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 311–324. https://doi.org/10.18653/v1/2021.naacl-main.29

[23] Diptikalyan Saha, Avrilia Floratou, Karthik Sankaranarayanan, Umar Farooq Minhas, Ashish R. Mittal, and Fatma Özcan. 2016. ATHENA: an ontology-driven system for natural language querying over relational data stores. *Proc. VLDB Endow.* 9, 12 (Aug. 2016), 1209–1220. https://doi.org/10.14778/2994509.2994536

[24] Mateus Santos Saldanha and Luciano Antonio Digiampietri. 2024. ChatGPT and Bard Performance on the POSCOMP Exam. In *Proceedings of the 20th Brazilian Symposium on Information Systems* (Juiz de Fora, Brazil) *(SBSI '24)*. Association for Computing Machinery, New York, NY, USA, Article 49, 10 pages. https://doi.org/10.1145/3658271.3658320

[25] Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. 2022. XRICL: Cross-lingual Retrieval-Augmented In-Context Learning for Cross-lingual Text-to-SQL Semantic Parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 5248–5259. https://doi.org/10.18653/v1/2022.findings-emnlp.384

[26] Qwen Team. 2024. Qwen2.5: A Party of Foundation Models. https://qwenlm.github.io/blog/qwen2.5/

[27] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL] https://arxiv.org/abs/2302.13971

[28] Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 7567–7578. https://doi.org/10.18653/v1/2020.acl-main.677

[29] Xiaojun Xu, Chang Liu, and Dawn Song. 2017. SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning. arXiv:1711.04436 [cs.CL] https://arxiv.org/abs/1711.04436

[30] An Yang and et al. 2024. Qwen2 Technical Report. arXiv:2407.10671 [cs.CL] https://arxiv.org/abs/2407.10671

[31] Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. 2021. GraPPa: Grammar-Augmented Pre-Training for Table Semantic Parsing. arXiv:2009.13845 [cs.CL] https://arxiv.org/abs/2009.13845

[32] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 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*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 3911–3921. https://doi.org/10.18653/v1/D18-1425

[33] Bin Zhang, Yuxiao Ye, Guoqing Du, Xiaoru Hu, Zhishuai Li, Sun Yang, Chi Harold Liu, Rui Zhao, Ziyue Li, and Hangyu Mao. 2024. Benchmarking the Text-to-SQL Capability of Large Language Models: A Comprehensive Evaluation. arXiv:2403.02951 [cs.CL] https://arxiv.org/abs/2403.02951

[34] Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. arXiv:1709.00103 [cs.CL] https://arxiv.org/abs/1709.00103

[35] Xiaohu Zhu, Qian Li, Lizhen Cui, and Yongkang Liu. 2024. Large Language Model Enhanced Text-to-SQL Generation: A Survey. arXiv:2410.06011 [cs.DB] https://arxiv.org/abs/2410.06011

Received 18 November 2024; revised 12 March 2024; accepted 5 June 2024