DataCNPJ: Um Conjunto de Dados Público para Avaliação de *Text-to-SQL* e *Schema-Linking* em Inglês e em Português

Paulo H. C. Silva[1], **Letícia O. Silva**[1], **Érick V. Issa**[1], **Fabrício A. Silva**[1]

[1]Laboratório de Inteligência em Sistemas Pervasivos e Distribuídos (NESPeD-Lab)
Universidade Federal de Viçosa - Campus Florestal (UFV-CAF)

{paulo.h.carneiro, leticia.silva1, erick.issa, fabricio.asilva}@ufv.br

Abstract. *The main benchmarks used to evaluate the performance of language models on the Text-to-SQL task, such as Spider and BIRD, are limited to the English language, do not fully reflect real-world scenarios, and do not provide explicit support for evaluating the Schema-Linking subtask. DataCNPJ, a dataset built from public data from the Brazilian National Registry of Legal Entities (CNPJ), was designed for the evaluation of models on Text-to-SQL and Schema-Linking tasks, providing multilingual support (Portuguese and English), 187 queries with different levels of complexity, and a validation scenario based on real-world data.*

Resumo. *Os principais benchmarks utilizados para validação do desempenho de modelos de linguagem na tarefa de Text-to-SQL, como Spider e BIRD, são limitados ao idioma inglês, não refletem plenamente cenários reais e não oferecem suporte explícito para avaliação da subtarefa de Schema-Linking. O conjunto de dados DataCNPJ, construído a partir de dados públicos do Cadastro Nacional da Pessoa Jurídica (CNPJ), foi projetado para avaliação de modelos nas tarefas de Text-to-SQL e Schema-Linking, oferecendo suporte multilíngue (português e inglês), 187 consultas com diferentes níveis de complexidade e um cenário de validação baseado em dados reais.*

1. Introdução

A tarefa de *Text-to-SQL*, que consiste em mapear linguagem natural em consultas SQL, tem ganhado destaque no estado da arte, impulsionada pelo avanço dos grandes modelos de linguagem. Essa tarefa possui um impacto relevante ao democratizar o acesso a dados armazenados em bancos de dados relacionais, permitindo que usuários sem conhecimento técnico em SQL possam acessar os dados por meio de linguagem natural.

Conjuntos de dados para treinamento e avaliação da tarefa de *Text-to-SQL* são fundamentais para o avanço da área. *Benchmarks* amplamente utilizados, como Spider [Yu et al. 2018] e BIRD [Li et al. 2023], têm sido adotados na literatura. No entanto, apesar de abrangerem múltiplos domínios, esses conjuntos não representam cenários reais e são restritos ao idioma inglês.

Outro aspecto relevante é a decomposição da tarefa de *Text-to-SQL* em subtarefas como o *Schema-Linking* [Pourreza and Rafiei 2024, Silva et al. 2025b, Silva et al. 2025a], responsável por identificar os componentes relevantes do esquema do banco de dados para a elaboração da consulta SQL. No entanto, *benchmarks* de *Text-to-SQL*, como Spider [Yu et al. 2018] e BIRD [Li et al. 2023], não disponibilizam anotações explícitas que permitam a avaliação direta do desempenho dos modelos nessa subtarefa.

Diante dessas limitações, este trabalho apresenta o DataCNPJ[1], um conjunto de dados construído a partir de dados públicos do Cadastro Nacional da Pessoa Jurídica (CNPJ), disponibilizados pela Receita Federal do Brasil. O *dataset* foi projetado para apoiar a avaliação de modelos de linguagem em tarefas de *Text-to-SQL* e *Schema-Linking* em um cenário real, possibilitando experimentos tanto em inglês quanto em português.

O DataCNPJ inclui: (i) um banco de dados relacional estruturado; (ii) consultas em linguagem natural em português e inglês; (iii) consultas SQL correspondentes; e (iv) anotações estruturadas para *Schema-Linking*. Além disso, as consultas foram classificadas por níveis de complexidade, permitindo análises mais detalhadas do desempenho dos modelos. Dessa forma, o DataCNPJ contribui para o avanço da área ao oferecer um cenário realista, multilíngue e reprodutível para avaliação de modelos de *Text-to-SQL* e tarefas relacionadas.

2. Trabalhos Relacionados

A tarefa de *Text-to-SQL* tem sido amplamente estudada nos últimos anos [Floratou et al. 2024, Hong et al. 2024, Liu et al. 2024], impulsionada pela relevância da linguagem SQL e pelo crescente interesse em democratizar o acesso a dados armazenados em bancos relacionais. Esse progresso é impulsionado tanto pelos avanços no desempenho de modelos de linguagem quanto pela disponibilidade de conjuntos de dados adequados para treinamento e avaliação.

O WikiSQL [Zhong et al. 2017] foi um dos primeiros conjuntos de dados de larga escala, focado em consultas simples sobre tabelas individuais. Posteriormente, o Spider [Yu et al. 2018] introduziu consultas mais complexas envolvendo múltiplas tabelas e diferentes domínios, além de uma divisão entre bases de treino e teste que permite avaliar a capacidade de generalização dos modelos. A partir desse *benchmarks*, diversas extensões e variações foram propostas, incluindo versões sintéticas, como o Spider-Syn[Gan et al. 2021a], cenários com maior realismo e ruído, como o Spider-DK[Gan et al. 2021b], e versões mais recentes voltadas a desafios mais complexos, como o Spider 2.0[Lei et al. 2025]. Outro *benchmark* proposto com o objetivo de aproximar a avaliação de cenários mais realistas é o BIRD [Li et al. 2023], que inclui bancos de dados maiores e consultas mais desafiadoras.

Alguns trabalhos têm explorado alternativas para cenários mais próximos de aplicações reais, como o estudo apresentado em [Oliveira et al. 2024], que utiliza o banco de dados Mondial[May 1999] como uma aproximação de bases reais, com esquemas mais extensos e maior complexidade estrutural. Apesar desses avanços, todos os *benchmarks* citados são em inglês e não fornecem anotações explícitas que permitam a avaliação isolada da tarefa de *Schema-Linking*. Nesse contexto, o DataCNPJ busca preencher essas lacunas ao oferecer um conjunto de dados baseado em dados públicos reais, com suporte multilíngue (português e inglês) e anotações explícitas para a tarefa de *Schema-Linking*.

Dessa forma, o DataCNPJ complementa os *benchmarks* existentes ao fornecer um cenário realista, reprodutível e voltado para aplicações práticas, posicionando-se como um conjunto de dados voltado para avaliação e comparação de modelos de linguagem no desempenho da tarefa de *Text-to-SQL* e de *Schema-Linking*.

[1]https://huggingface.co/datasets/NESPED-GEN/DataCNPJ

3. Construção do *Dataset*

3.1. Construção do Banco de Dados

A tarefa de *Text-to-SQL* possui como requisito fundamental a existência de um banco de dados relacional no qual as consultas SQL possam ser executadas. Para atender esse requisito, construímos um banco de dados relacional no formato SQLite a partir dos dados públicos[2] do Cadastro Nacional da Pessoa Jurídica (CNPJ), disponibilizados pela Receita Federal do Brasil.

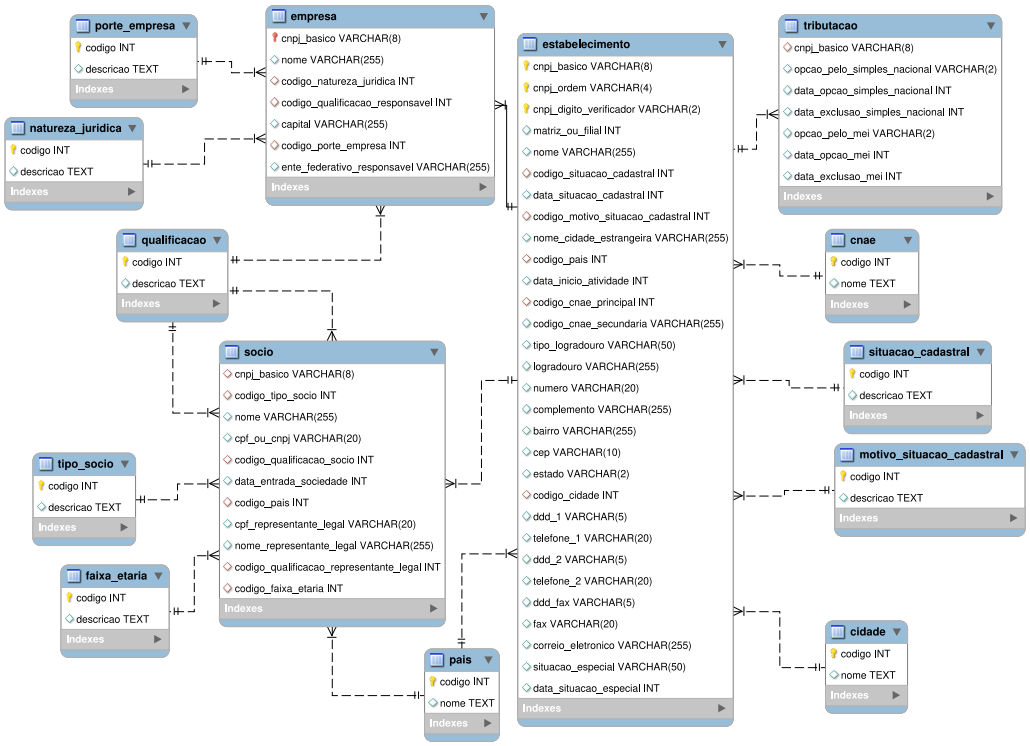

Figura 1. Diagrama EER do Banco de Dados (versão em português)

Os dados consistem em informações detalhadas sobre empresas registradas no Brasil e são disponibilizados em arquivos compactados organizados por mês e ano de referência. Cada conjunto mensal contém múltiplos arquivos no formato CSV, nomeados de acordo com o tipo de informação armazenada, como *Empresas*, *Estabelecimentos*, *Sócios*, *CNAEs*, *Municípios*, *Países*, *Naturezas Jurídicas*, *Qualificações*, *Motivos* e *Simples*. Devido ao grande volume de dados, algumas dessas categorias são distribuídas em múltiplos arquivos numerados.

A interpretação adequada dos dados depende do dicionário[3] disponibilizado pela Receita Federal, que descreve os atributos e os valores codificados presentes nos arquivos. Por exemplo, o atributo `PORTE_EMPRESA` armazena códigos numéricos que representam categorias como microempresa ou empresa de pequeno porte. Essas informações foram utilizadas para enriquecer o banco de dados relacional, por meio da criação de tabelas auxiliares contendo a descrição semântica desses códigos.

[2]https://dados.gov.br/dados/conjuntos-dados/cadastro-nacional-da-pessoa-juridica—cnpj
[3]https://www.gov.br/receitafederal/dados/cnpj-metadados.pdf

O código utilizado para a extração e transformação dos dados está disponível em um repositório GitHub[4], desenvolvido a partir de adaptações e atualizações de *scripts* previamente disponibilizados no GitHub[5].

Após a etapa de extração, foi realizado um processo de curadoria dos dados, incluindo: (i) organização do esquema relacional; (ii) padronização e nomeação das tabelas e atributos; (iii) criação de tabelas adicionais para representar domínios categóricos; e (iv) definição de chaves e relacionamentos, garantindo a integridade referencial. Como resultado, foram geradas duas versões do banco de dados: uma em português e outra em inglês, com esquemas equivalentes. A Figura 1 apresenta a estrutura do banco de dados na versão em português. O banco final contém 14 tabelas e 75 colunas. Os registros são do período de abril de 2026 e contém dados de 4.494.860 empresas e 4.660.147 estabelecimentos brasileiros. Os arquivos dos bancos no formato SQLite estão disponíveis no repositório do projeto[6].

3.2. Elaboração de Consultas em Linguagem Natural e SQL

A partir da construção do banco de dados relacional, foi realizada a etapa de elaboração das consultas em linguagem natural e das respectivas consultas SQL de referência. A consulta SQL de referência representa uma solução correta que, ao ser executada no banco de dados, responde adequadamente à consulta em linguagem natural.

No contexto da tarefa de *Text-to-SQL*, modelos de linguagem recebem como entrada uma consulta em linguagem natural juntamente com o esquema do banco de dados, e devem gerar uma consulta SQL correspondente. Dessa forma, as consultas SQL de referência podem ser utilizadas para treinamento adicional do modelo e para validação do desempenho dos modelos, que é o principal objetivo do conjunto de dados proposto neste trabalho, por meio da comparação entre a consulta gerada e a consulta esperada.

A avaliação do desempenho em *Text-to-SQL* pode ser realizada por meio de diferentes métricas. A *Exact Match Accuracy* (EM) verifica se a consulta gerada corresponde exatamente à consulta de referência, comparando seus componentes estruturais, sem considerar valores literais. Já a *Execution Accuracy* (EX) avalia se ambas as consultas produzem o mesmo resultado quando executadas no banco de dados [Wang et al. 2025].

Ambas as métricas podem ser calculadas utilizando a ferramenta *Test Suite Accuracy*[7], proposta em [Zhong et al. 2020], que também permite a análise segmentada por níveis de dificuldade (*easy*, *medium*, *hard* e *extra hard*), conforme a taxonomia introduzida no Spider [Yu et al. 2018].

A ferramenta utiliza um *parser* para o processamento e a classificação das consultas, o qual apresenta algumas restrições sintáticas. Dessa forma, as consultas foram elaboradas de modo a garantir compatibilidade com o *parser*. Os arquivos necessários para a utilização do *Test Suite Accuracy* com o DataCNPJ estão disponíveis no repositório do projeto.

[4]https://github.com/ErickIssa/Receita_Federal_do_Brasil_-_Dados_Publicos_CNPJ
[5]https://github.com/aphonsoar/Receita_Federal_do_Brasil_-_Dados_Publicos_CNPJ
[6]https://github.com/lleticiasilvaa/DataCNPJ
[7]https://github.com/taoyds/test-suite-sql-eval

Inicialmente, foram elaborados manualmente 65 pares de perguntas em linguagem natural, em inglês, juntamente com suas respectivas consultas SQL executáveis na versão do banco de dados em inglês. Para expandir o conjunto de dados, foram utilizados os modelos de linguagem `Gemini-2.5-pro` e `Gemini-2.5-flash` para a geração de novos pares de consultas. No total, foram geradas 300 novas consultas. O código e os *prompts* utilizados estão disponíveis no repositório do projeto.

O conjunto de consultas sintéticas foi revisado manualmente com o objetivo de remover duplicatas e adequar as consultas às restrições do *Test Suite Accuracy*. Adicionalmente, as consultas foram executadas no banco de dados para identificar e corrigir erros de sintaxe. Ao final desse processo, 122 consultas sintéticas foram incorporadas, totalizando 187 pares de consultas em inglês. A complexidade das consultas foi determinada por meio do script disponibilizado pelo *Test Suite Accuracy*, e a Tabela 1 apresenta a distribuição das consultas por nível de dificuldade.

Tabela 1. Distribuição do Nível de Dificuldade das Consulas SQL de Referência

	easy	*medium*	*hard*	*extra hard*	Total
Não-Sintéticas	21 (32%)	24 (37%)	9 (14%)	11 (17%)	65 (100%)
Geral	27 (14%)	44 (24%)	28 (15%)	88 (47%)	187 (100%)

Uma versão preliminar deste conjunto de dados, composta por um subconjunto distinto (com algumas interseções) contendo 200 consultas em inglês, foi utilizada em experimentos apresentados em [Silva et al. 2025b] e [Silva et al. 2025a]. Os códigos desses experimentos foram disponibilizados pelos respectivos trabalhos e podem ser utilizados como referência para o uso do dataset na validação das tarefas de *Text-to-SQL* e *Schema-Linking*.

3.3. Tradução para Português

Com o objetivo de disponibilizar o conjunto de dados para viabilizar a avaliação de modelos de linguagem na tarefa de *Text-to-SQL* utilizando consultas e esquemas em português, foi realizada a tradução das consultas em linguagem natural originalmente elaboradas em inglês. O modelo `unicamp-dl/translation-en-pt-t5`[8] [Lopes et al. 2020] foi utilizado nessa etapa, seguido de uma revisão manual para correção de possíveis inconsistências e erros gramaticais introduzidos pelo processo automático de tradução.

A tradução das consultas SQL foi realizada por meio de um script em Python, disponível no repositório do projeto, baseado em um mapeamento entre os nomes de tabelas e colunas das versões em inglês e em português do banco de dados. A Figura 2 apresenta um exemplo desse mapeamento entre as tabelas. Esse processo consistiu na substituição dos identificadores em inglês identificados na *string* por seus correspondentes em português, preservando-se as palavras-chave da linguagem SQL. Como etapa de validação, todas as consultas traduzidas foram executadas no banco de dados em português, garantindo sua correção sintática, bem como a equivalência dos resultados em relação às consultas em inglês.

[8]https://huggingface.co/unicamp-dl/translation-en-pt-t5

```
'company': 'empresa'
'establishment': 'estabelecimento'
'partner': 'socio'
'taxation': 'tributacao'
'registration_status_reason': 'motivo_situacao_cadastral'
'city': 'cidade'
'country': 'pais'
'legal_nature': 'natureza_juridica'
'qualification': 'qualificacao'
'company_size': 'porte_empresa'
'registration_status': 'situacao_cadastral'
'partner_type': 'tipo_socio'
'age_range': 'faixa_etaria'
```

Figura 2. Mapeamento entre as Tabelas do esquema em Inglês e Português

3.4. Extração do *Schema-Linking*

Além de disponibilizar um conjunto de dados em português para validação da tarefa de *Text-to-SQL*, este trabalho também contribui com anotações explícitas para a tarefa de *Schema-Linking*. Essa tarefa consiste na identificação das tabelas e colunas relevantes do banco de dados para a elaboração da consulta SQL.

No cenário tradicional de *Text-to-SQL*, modelos de linguagem recebem como entrada a consulta em linguagem natural juntamente com o esquema completo do banco de dados. Embora o *Schema-Linking* esteja implícito nesse processo, ele pode ser tratado como uma etapa independente e prévia no pipeline de geração de consultas SQL, como explorado em trabalhos anteriores [Pourreza and Rafiei 2024, Silva et al. 2025b, Silva et al. 2025a]. Nesse contexto, a identificação prévia dos elementos relevantes do esquema permite reduzir o espaço do modelo, tornando a geração da consulta SQL menos complexa. Ao filtrar o esquema e fornecer apenas as tabelas e colunas relevantes como contexto, o modelo pode concentrar seus esforços na estrutura da consulta SQL [Yang et al. 2024, Volvovsky et al. 2024].

Diante disso, tratar o *Schema-Linking* como uma tarefa independente demanda a existência de conjuntos de dados que permitam avaliar diretamente o desempenho de modelos nessa etapa. Neste trabalho, adotamos uma representação estruturada para o *Schema-Linking* no formato JSON, na qual as tabelas relevantes são utilizadas como chaves, e cada tabela está associada a um conjunto de colunas relevantes. Essa abordagem também foi utilizada em [Silva et al. 2025b].

Para a elaboração das anotações de *Schema-Linking*, foi utilizada a biblioteca `sql-metadata`[9], que extrai automaticamente as tabelas referenciadas em uma consulta SQL. Para a identificação das colunas, foi adotada uma estratégia baseada na correspondência direta entre os nomes das colunas presentes na consulta SQL de referência e os atributos das tabelas selecionadas. A Figura 3 exemplifica um JSON com o *Schema-Linking* esperado juntamente com uma comparação entre o esquema reduzido a tabelas e colunas e o esquema completo. Essa visualização destaca o impacto da filtragem do esquema.

[9]https://pypi.org/project/sql_metadata/

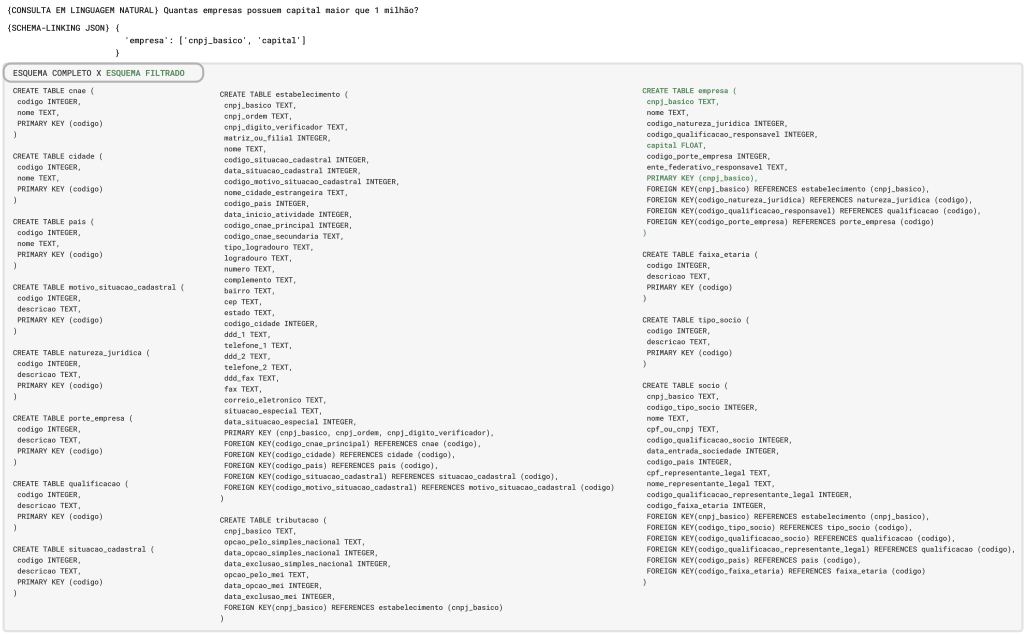

Figura 3. Exemplo de *Schema-Linking* e do impacto de filtrar o esquema

4. DataCNPJ

O DataCNPJ é um conjunto de dados projetado para avaliação de modelos de linguagem nas tarefas de *Text-to-SQL* e *Schema-Linking*. O *dataset* contém 187 consultas, sendo 65 elaboradas manualmente e as demais geradas de forma sintética. Cada consulta está disponível em português e em inglês, permitindo sua execução nos respectivos bancos de dados. Os esquemas dos bancos diferem apenas na nomenclatura de tabelas e colunas (em português e inglês), enquanto os dados armazenados permanecem em português. Dessa forma, são disponibilizadas duas versões equivalentes do banco de dados, cada uma contendo 14 tabelas e 75 colunas.

Cada instância do *dataset* é composta por um identificador numérico único (`question_id`), pela consulta em linguagem natural em inglês (`question_EN`) e sua respectiva consulta SQL para o banco em inglês (`query_cnpjEN`), além da anotação de *Schema-Linking* em formato JSON correspondente (`schema_linking_cnpjEN`). O conjunto também disponibiliza a versão em português da consulta em linguagem natural (`question_PT`), sua respectiva consulta SQL para o banco em português (`query_cnpjPT`) e a anotação de *Schema-Linking* correspondente (`schema_linking_cnpjPT`). Adicionalmente, cada instância possui um atributo booleano indicando se a consulta foi gerada sinteticamente (`synthetic`) e um atributo referente ao nível de complexidade da consulta (`hardness`), classificado em *easy*, *medium*, *hard* ou *extra*, sendo esta última categoria correspondente ao nível *extra hard*.

O conjunto de dados está publicamente disponível na plataforma Hugging Face[10]. Os códigos utilizados para extração, transformação e construção do banco de dados[11], bem como para geração e processamento das consultas[12], estão disponíveis no GitHub.

[10]https://huggingface.co/datasets/NESPED-GEN/DataCNPJ
[11]https://github.com/ErickIssa/Receita_Federal_do_Brasil_-_Dados_Publicos_CNPJ
[12]https://github.com/lleticiasilvaa/DataCNPJ

5. Conclusão e Trabalhos Futuros

Este trabalho apresentou o DataCNPJ, um conjunto de dados voltado para a avaliação de modelos de linguagem nas tarefas de *Text-to-SQL* e *Schema-Linking*. Diferentemente dos principais *benchmarks* existentes, o DataCNPJ é construído a partir de dados públicos reais, oferece suporte multilíngue (inglês e português) e inclui anotações explícitas para a avaliação da subtarefa de *Schema-Linking*.

O processo de construção envolveu a criação de um banco de dados relacional, a elaboração e validação de consultas em linguagem natural e SQL, a geração de exemplos sintéticos e a tradução sistemática para o português. Além disso, o conjunto de dados foi estruturado para ser compatível com a ferramenta *Test Suite Accuracy*, facilitando o cálculo das métricas de *Execution Accuracy* (EX) e *Exact Match Accuracy* (EM).

Como trabalhos futuros, pretende-se ampliar a quantidade de consultas elaboradas manualmente e validar as consultas com especialistas, como profissionais do domínio fiscal. Também pretende-se expandir a quantidade de consultas sintéticas, explorando diferentes modelos de linguagem para sua geração e buscando balancear a distribuição entre os diferentes níveis de complexidade. Por fim, pretende-se explorar diferentes estratégias de nomeação para tabelas e colunas do banco de dados, bem como múltiplas formulações em linguagem natural para uma mesma consulta SQL, permitindo investigar o impacto dessas variações no desempenho de modelos de linguagem.

Uso de Inteligência Artificial

Ferramentas de IA generativa foram utilizadas neste trabalho para auxiliar na geração de consultas sintéticas em linguagem natural e SQL, na tradução de consultas do inglês para o português e na revisão textual do artigo.

Referências

Floratou, A., Psallidas, F., Zhao, F., Deep, S., Hagleither, G., Tan, W., Cahoon, J., Alotaibi, R., Henkel, J., Singla, A., Grootel, A. V., Chow, B., Deng, K., Lin, K., Campos, M., Emani, K. V., Pandit, V., Shnayder, V., Wang, W., and Curino, C. (2024). NL2SQL is a solved problem... Not! In *Conference on Innovative Data Systems Research*.

Gan, Y., Chen, X., Huang, Q., Purver, M., Woodward, J. R., Xie, J., and Huang, P. (2021a). Towards Robustness of Text-to-SQL Models against Synonym Substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 2505–2515, Online. Association for Computational Linguistics.

Gan, Y., Chen, X., and Purver, M. (2021b). Exploring Underexplored Limitations of Cross-Domain Text-to-SQL Generalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8926–8931, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hong, Z., Yuan, Z., Zhang, Q., Chen, H., Dong, J., Huang, F., and Huang, X. (2024). Next-Generation Database Interfaces: A Survey of LLM-based Text-to-SQL. *ArXiv*, abs/2406.08426.

Lei, F., Chen, J., Ye, Y., Cao, R., Shin, D., Su, H., Suo, Z., Gao, H., Hu, W., Yin, P., Zhong, V., Xiong, C., Sun, R., Liu, Q., Wang, S., and Yu, T. (2025). Spider 2.0: Evaluating Language Models on Real-World Enterprise Text-to-SQL Workflows. *ArXiv*, abs/2411.07763.

Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., Wang, B., Qin, B., Geng, R., Huo, N., Zhou, X., Ma, C., Li, G., Chang, K. C., Huang, F., Cheng, R., and Li, Y. (2023). Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQLs. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Liu, X., Shen, S., Li, B., Ma, P., Jiang, R., Zhang, Y., Fan, J., Li, G., Tang, N., and Luo, Y. (2024). A Survey of NL2SQL with Large Language Models: Where are we, and where are we going? *ArXiv*, abs/2408.05109.

Lopes, A., Nogueira, R., Lotufo, R., and Pedrini, H. (2020). Lite Training Strategies for Portuguese-English and English-Portuguese Translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 833–840, Online. Association for Computational Linguistics.

May, W. (1999). Information Extraction and Integration with FLORID: The MONDIAL Case Study. Technical Report 131, Universität Freiburg, Institut für Informatik. Available from `http://dbis.informatik.uni-goettingen.de/Mondial`.

Oliveira, A., Nascimento, E., Pinheiro, J. a., Avila, C. V. S., Coelho, G., Feijó, L., Izquierdo, Y., García, G., Leme, L. A. P. P., Lemos, M., and Casanova, M. A. (2024). Small, Medium, and Large Language Models for Text-to-SQL. In *Conceptual Modeling: 43rd International Conference, ER 2024, Pittsburgh, PA, USA, October 28–31, 2024, Proceedings*, page 276–294, Berlin, Heidelberg. Springer-Verlag.

Pourreza, M. and Rafiei, D. (2024). DTS-SQL: Decomposed Text-to-SQL with Small Large Language Models. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8212–8220, Miami, Florida, USA. Association for Computational Linguistics.

Silva, L., Silva, P. H., and Silva, F. (2025a). Leis de Escala para Text-to-SQL: Um Estudo sobre a Relação entre Tamanho e Desempenho de Modelos de Linguagem. In *Anais do XL Simpósio Brasileiro de Bancos de Dados*, pages 140–153, Porto Alegre, RS, Brasil. SBC.

Silva, P., Silva, L., and Silva, F. (2025b). Towards Small Language Models for Text-to-SQL. In *Anais da XXXV Brazilian Conference on Intelligent Systems*, pages 516–531, Porto Alegre, RS, Brasil. SBC.

Volvovsky, S., Marcassa, M., and Panbiharwala, M. (2024). DFIN-SQL: Integrating Focused Schema with DIN-SQL for Superior Accuracy in Large-Scale Databases. *ArXiv*, abs/2403.00872.

Wang, B., Ren, C., Yang, J., Liang, X., Bai, J., Chai, L., Yan, Z., Zhang, Q., Yin, D., Sun, X., and Li, Z. (2025). MAC-SQL: A Multi-Agent Collaborative Framework for Text-to-SQL. In *Proceedings of the 31st International Conference on Computational*

Linguistics, COLING 2025, pages 540–557, Abu Dhabi, UAE. Association for Computational Linguistics.

Yang, S., Su, Q., Li, Z., Li, Z., Mao, H., Liu, C., and Zhao, R. (2024). SQL-to-Schema Enhances Schema Linking in Text-to-SQL. In *Database and Expert Systems Applications - 35th International Conference, DEXA 2024*.

Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., and Radev, D. (2018). Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMLP 2018*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Zhong, R., Yu, T., and Klein, D. (2020). Semantic Evaluation for Text-to-SQL with Distilled Test Suites. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*.

Zhong, V., Xiong, C., and Socher, R. (2017). Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *ArXiv*, abs/1709.00103.