Automatização de Consultas SQL em Bancos de Dados Utilizando Modelos de Linguagem Natural

Thiago Pádua de Carvalho
2020007066

Pesquisa Científica

Orientador:
Adriano Alonso Veloso

Universidade Federal de Minas Gerais
Novembro/2024

1 Introdução

Vive-se atualmente uma tendência em que o volume de dados gerado por empresas, governos e indivíduos alcançou níveis sem precedentes. Esse crescimento reflete a digitalização de processos, a expansão de serviços online e a conectividade global. Em paralelo, decisões baseadas em dados tornaram-se cruciais para diferentes tipos de serviços, permeando as mais variadas áreas do conhecimento e impactando o modo a partir do qual se estrutura toda a inteligência organizacional.

Podemos entender os dados como matéria prima lógica que será consumida em incontáveis aplicações, permitindo melhoria de performance, redução de custos, ganho de flexibilidade operacional, planejamento respaldado em conhecimentos empíricos, dentre diversos outros. Sendo assim, há uma mudança de paradigma que afeta a maneira com a qual os profissionais gerais se relacionam com dados, exigindo por muitas vezes um conhecimento técnico mais avançado para lidar com as aplicações utilizadas. De outro lado, há uma demanda dos usuários por obtenção rápida, intuitiva e assertiva de informações dos sistemas.

Nesse sentido, Text-to-SQL surge como uma alternativa para facilitar o acesso aos dados. Ele consiste em um método de transformar perguntas feitas em linguagem natural para as queries correspondentes em SQL. Sendo assim, é possível reduzir a expertise técnica necessária para lidar com os dados e tornar a experiência de obtenção de informações mais fluida e simples para os usuários. Além disso, essa técnica aumenta a eficiência do processamento de dados e contribui com um amplo leque de aplicações como serviços inteligentes de bancos de dados (BD), análise automática de dados e aplicações de perguntas e respostas em BDs.

É possível também ressaltar que Text-to-SQL tem um grande potencial para aumento de produtividade principalmente no que diz respeito à economia de tempo, desde a capacitação dos profissionais até a própria elaboração das consultas sobre os dados disponíveis. Sua versatilidade de aplicações ainda é outra vantagem. Como desvantagens pode-se citar o alto teor de erro presente no atual estado da arte das soluções existentes e o elevado investimento inicial, que no entanto é compensado em médio prazo.

O objetivo geral do projeto é desenvolver uma aplicação própria de Text-to-SQL, capaz de traduzir perguntas em linguagem natural em consultas SQL de modo otimizado. Para isso, pretende-se explorar modelos de linguagem avançados, investigando sua capacidade de gerar consultas a partir de entradas textuais obtidas em um conjunto de dados relevante com aplicações principalmente em saúde. Para mais, o projeto visa não apenas apresentar uma solução funcional, mas também oferecer insights que contribuam para o avanço de tecnologias Text-to-SQL em contextos práticos e acessíveis.

Ao final do POC II, o objetivo é consolidar um protótipo que permita a avaliação da qualidade dos resultados obtidos ao transformar um texto livre em cláusulas SQL. Isso será possível através da implementação de modelos de Text-to-SQL feita com o intermédio de Large Language Models (LLMs) [3], que serão comparados quanto à sua assertividade dados os diferentes parâmetros de construção e também suas arquiteturas e representações.

Para o POC I, o objetivo é consolidar pelo menos um modelo capaz de realizar a tarefa de Text-to-SQL utilizando uma LLM de código aberto, construindo o conhecimento necessário para entender seus aspectos arquiteturais e também avaliar a primeira solução para então buscar pontos de melhora que serão desenvolvidos mais profundamente na próxima etapa.

2 Referencial Teórico

O método de Text-to-SQL envolve uma série de conhecimentos que interessam tanto os profissionais de bancos de dados quanto aqueles envolvidos com processamento de linguagem natural. Este é um problema que tem natureza multidisciplinar e engloba uma série de processos que devem trabalhar em conjunto para o bom funcionamento geral. O desenho de uma proposta de texto em língua natural para SQL envolve delicados parâmetros e necessita de um ajuste cuidadoso para atingir resultados satisfatórios. Com isso em vista, a seguir são apresentados alguns conceitos fundamentais para o assunto e também um panorama do atual estado da arte.

2.1 Structured Query Language (SQL)

SQL é uma linguagem de programação para armazenar e processar informações em um banco de dados relacional[2]. Um BD relacional, por sua vez, guarda dados através de tabelas que podem representar entidades ou relacionamentos próprios do conjunto de conhecimentos armazenados. Cada linha nas tabelas representa entradas únicas que estão sendo salvas. As instruções SQL operam sobre o banco de dados e são capazes de criar, armazenar, alterar, deletar e especialmente consultar as entradas, que é o tema de nosso maior interesse.

```sql
-- 1. Cria de uma tabela simples chamada "Clientes"
CREATE TABLE Clientes (
    id INT PRIMARY KEY,          -- Identificador
    nome VARCHAR(50),
    email VARCHAR(50),
    cidade VARCHAR(50)
);

-- 2. Insere dados na tabela
INSERT INTO Clientes (id, nome, email, cidade) VALUES
(1, 'Ana Silva', 'ana.silva@email.com', 'Sao Paulo'),
(2, 'Carlos Pereira', 'carlos.pereira@email.com', 'Rio de Janeiro'
   ),
(3, 'Mariana Souza', 'mariana.souza@email.com', 'Belo Horizonte');

-- 3. Consulta
SELECT nome, email
FROM Clientes
WHERE cidade = 'Sao Paulo';
```

Listing 1: Exemplo de criação, inserção e consulta em SQL.

O grande foco do trabalho será obter queries no formato de SELECT, como mostrado na imagem acima a partir de linguagem natural. Como exemplo, a consulta vista poderia vir de uma pergunta como:

"Qual é o nome e o email dos clientes moradores de São Paulo?"

Naturalmente, a consulta por linguagem natural é mais acessível e facilita que usuários não experientes tenham acesso aos dados.

2.2 Large Language Models

Large Language Model é um arquétipo de inteligência artificial treinado com enormes volumes de dados textuais e capaz de entender e gerar linguagem natural de forma sofisticada, sendo apto a realizar tarefas dos mais variados tipos. As LLMs - devido à sua natureza capaz de interpretar muito bem a linguagem humana - mudaram o paradigma com o que se faz Text-to-SQL. Modelos cada vez mais avançados, como o GPT-4[4] surgem e revolucionam diversas aplicações. Eles conseguem identificar palavras-chave, compreender a estrutura do banco de dados e gerar consultas que correspondam à intenção do usuário. Sendo assim, constituem uma parte fundamental do processo obtenção de queries SQL e por isso grande parte do esforço da aplicação se concentra em ajustar os inputs fornecidos para os modelos e também em fazer seu processo de tuning - visto adiante.

Por funcionarem como caixas pretas, é preciso fazer uma análise cuidadosa de como fornecer entradas para as LLMs. O processo de ajustar o formato do input para otimizar os resultados obtidos é chamado de prompt engineering. É necessário trabalhar na representação das questões feitas, isto é, na edição, indicação de comandos com o auxílio de símbolos e elaboração do padrão de entrada; na disponibilidade (in context) ou não (zero-shot) de exemplos e no formato geral que vai ser apresentado ao modelo.

Finalmente, o processo de fine-tuning, é uma técnica que maximiza os resultados obtidos pela LLM através de treinamento supervisionado direcionado à tarefa de Text-to-SQL. Ele faz o alinhamento do modelo, especializando-o e evitando outputs enviesados e também alucinações.

Em Text-to-SQL a pergunta passa por um pré-processamento e é representada de acordo com um dos diversos modos definidos - com exemplos gerados - para então servir de input à LLM que, também previamente, passou por um processo anterior de fine-tuning. Esse procedimento garante o melhor desempenho observado no estado da arte atual[1].

2.3 Estado da Arte Atual

Como a evolução e emprego amplo de LLMs é algo relativamente recente, as aplicações de Text-to-SQL estão em pleno processo de melhora nos benchmarks adotados como padrão[5]. Modelos avançados como o GPT-4, treinado com dezenas de bilhões de parâmetros, apresentam os melhores resultados na tarefa, com cerca de 86% de acurácia na execução da query gerada a partir da técnica de DAIL-SQL[1], a qual equilibra a capacidade de aprendizado da LLM com a eficiência de tokens (uso otimizado de tokens no prompt) a partir de uma seleção criteriosa e manipulação de exemplos. Esses, por sinal são tópicos essenciais para se tratar e avaliar as soluções encontradas, além da taxa direta de sucesso.

Todas as possíveis combinações possíveis para compor a arquitetura deixam um desafio pendente para a área de Text-to-SQL e ainda é necessário explorar o trade-off entre quantidade e qualidade dos exemplos fornecidos como entrada para a LLM. Além disso, nem sempre uma melhor performance é viável, devido ao elevado custo da solução.

Pode-se dizer que ainda há muito a se explorar no contexto de Text-to-SQL, especialmente com o uso de LLMs com código aberto. Sua capacidade ainda é limitada em questão ao número de parâmetros e elas não funcionam bem com fine-tuning, pois overfitam com os dados, ocasionando até mesmo em uma piora nos resultados observados.

Esse é um campo em aberto e em pleno progresso, com um potencial enorme de impacto na sociedade.

3 Metodologia

3.1 Revisão da Literatura e Análise Exploratória dos dados

Como mostrado anteriormente, o processo de execução de Text-to-SQL é complexo e envolve uma série de áreas relacionadas à computação. Sendo assim, inicialmente é necessário fazer uma revisão da literatura, através da leitura de artigos relacionados às técnicas aplicadas, como também soluções anteriores a fim de realizar uma avaliação dos métodos e também servir de inspiração para o modelo a ser construído. É fundamental solidificar a base de conhecimento para trabalhar consistentemente em uma própria.

Além disso esse é o momento em que será feita a análise exploratória dos dados (EDA) com fim de entender melhor o objeto de trabalho, descobrir padrões e características que poderão ser exploradas futuramente para prompt engineering e também para o processo de fine-tuning da LLM escolhida. Esse é um passo fundamental para descobrir "atalhos"nas informações e fazer com que o modelo execute com menos tempo e mais assertividade.

3.2 Design Inicial da solução e Análise

Uma vez tendo estabelecido a maioria dos conceitos teóricos, é necessário começar a pensar no design da primeira solução a ser apresentada para o POC I. Foi elucidado previamente que há grande complexidade na montagem de um modelo Text-to-SQL, uma vez que esta envolve uma série de decisões de projeto a serem consideradas para montar uma arquitetura robusta e que apresente bons resultados. É necessário decidir qual Large Language Model será utilizado, tal como o prompt engineering, com definição de cenário de exemplos, isto é, zero-shot ou in-context learning e, no último caso, quantos serão selecionados para fornecer de input juntamente ao texto em língua natural. Por fim, será necessário testar o fine-tuning.

É esperado que esse processo seja acompanhado de uma avaliação detalhada do impacto de cada um dos itens citados e que ele funcione para a prática e consolidação dos termos teóricos. A expectativa é que este seja o processo que mais demande tempo nesta primeira fase.

3.3 Implementação de um Protótipo

Uma vez que as análises inicias sejam feitas, um primeiro protótipo será implementado com o objetivo de estabelecer uma entrega e então firmar um modelo que apresente bons resultados. Ele deve avaliar na prática os conceitos discutidos e fornecer uma base para a realização de testes avançados posteriores e ajustes. Uma revisão completa dos conceitos e das decisões de projeto será feita.

4 Resultados Esperados

Ao final do trabalho espera-se agregar conhecimento de maneira a consolidar conceitos relacionados tanto a Structured Query Language quanto a Large Language Models. Especialmente a interação entre eles por meio do método de Text-to-SQL deve ser compreendida profundamente, levando a sua concretização por meio de um protótipo que permita a avaliação da qualidade dos resultados obtidos ao transformar linguagem natural em consultas com sintaxe SQL.

Além disso, esse modelo de Text-to-SQL deve ser aplicado a dados reais e será capaz de traduzir perguntas em linguagem natural, feitas por profissionais da saúde, em consultas SQL precisas, que extraiam informações relevantes de bancos de dados médicos. Com isso, espera-se ter um impacto positivo, com redução de tempo e consultas claras em um contexto de fundamental relevância social.

Como objetivo final, está estabelecido fazer uma análise completa da solução, com descrição de impacto das mais variadas decisões de projeto e correlação entre as queries observadas, parâmetros e dados fornecidos para obter uma noção de explicabilidade do modelo.

Para alcançar este resultado é necessário que ao final do POC I um modelo inicial de Text-to-SQL esteja implementado e analisado. Ele representa uma entrega inicial e a solidificação dos conceitos teóricos. Além disso, será a partir dele o aprofundamento e melhora da solução para o POC II.

5 Etapas e Cronograma

- 17/11 - entrega da proposta de POC
- 18/11 - 27/11 - Revisão da Literatura e EDA
- 28/11 - 01/01 - Design Inicial da Solução e Análise
- 02/01 - 20/01 - Implementação do Protótipo

6 Referências Bibliográficas

Referências

[1] Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, Jingren Zhou. *Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation.*

[2] Amazon Web Services. *O que é SQL?*. Disponível em: `https://aws.amazon.com/pt/what-is/sql/`. Acesso em: 13 nov. 2024.

[3] IBM. *Large Language Models (LLMs)*. Disponível em: `https://www.ibm.com/topics/large-language-models`. Acesso em: 15 nov. 2024.

[4] OpenAI. *GPT-4 Technical Report*. CoRR abs/2303.08774, 2023.

[5] LILY Group at Yale University. 2018. *Spider 1.0, Yale Semantic Parsing and Text-to-SQL Challenge.* https://yale-lily.github.io/spider.