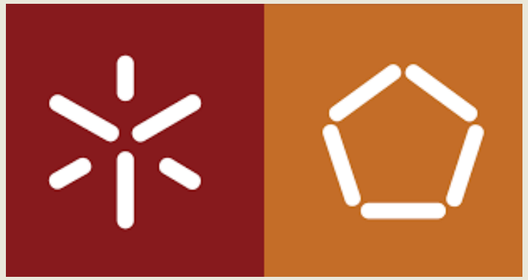


LLM-based Querying vs SQL

Eduardo Cunha PG55939
Jorge Rodrigues PG55966
Tiago Rodrigues PG56013
Vasco Faria PG57905

Introdução

Este trabalho analisa a evolução da interação de utilizadores com bases de dados relacionais, passando do uso tradicional de SQL para soluções baseadas em inteligência artificial, baseadas em linguagem natural

O trabalho propõe comparar diferentes abordagens utilizando benchmarks de sistemas de bases de dados consolidados, além de testes convencionais das ferramentas text-to-SQL.

Visão Geral

Text-to-SQL

OLTP vs OLAP

LLM

Benchmarks

Os benchmarks mais consolidados no tema do text-to-SQL são o BIRD e o SPIDER.

No entanto, o foco deste trabalho é incorporar e avaliar os três níveis com o TPC-C e o TPC-H, que são benchmarks consolidados e típicos de bases de dados, porque foram concebidos para avaliar diferentes dimensões do desempenho real de sistemas de gestão de bases de dados.

Métricas

Fizemos um levantamento das métricas mais frequentemente utilizadas em contextos de text-to-SQL, e estas são as que adotámos:

Exact Matching Accuracy (EM)

$$ExactMatchingAccuracy = \frac{1}{N}\sum_{i=1}^{N} I(Og_i = Ob_i)$$

Execution Accuracy (EX)

$$ExecutionAccuracy = \frac{1}{N}\sum_{i=1}^{N} I(f(Q_i, S_i) = A_i)$$

Valid Efficiency Score (VES)

$$ValidEfficiencyScore = \frac{1}{N}\sum_{i=1}^{N} I(f(Q_i, S_i) = A_i) * \frac{T^{gold}}{T^{gen}}$$

Structural Similarity (SS)

$$\mathbb{S}_i = \frac{\sum_{\{tin\ \text{Types}\}} \min\left(C_{i[t]}^{\text{ref}}, C_{i[t]}^{\text{gen}}\right)}{\sum_{\{tin\ \text{Types}\}} \max\left(C_{i[t]}^{\text{ref}}, C_{i[t]}^{\text{gen}}\right)}$$

Token F1 (F1)

$$F_1(c, A_r) = \max_{a \in A_r} \left(\frac{|t(a)|}{|t(a) \cap t(c)|} + \frac{|t(c)|}{|t(a) \cap t(c)|} \right)^{-1}$$

Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

Niveis de avaliação

1. Basic Text-to-SQL

2. Context-Aware SQL Generation

3. Direct Data Access

Basic Text-to-SQL

Q. What is the total amount spent on orders?

A.

```
SELECT
SUM(amount) AS
total_spent
FROM orders;
```

Basic Text-to-SQL

- **Prompt Engeenering**

Prompt:

Im a data engineer and want to know what is the total amount spent on orders by the customers of the service?

- **Iterative Refinement**

Prompt:
Im a data engineer and want to know what is the total amount spent on orders by the customers of the service?

Resposta:...

Prompt:
(Logs de erros da execução da resposta, se existirem)

Resposta:...

Prompt:
(Logs de erros da execução da resposta, se existirem)

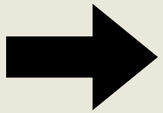
Resposta:...

Abordagem TPC-H

1. Executar queries originais TPC-H e guardar outputs de referência
2. Gerar queries usando LLMs, através das descrições NL
3. Executar queries geradas pelas LLMs e guardar os respetivos outputs.
4. Comparar resultados usando a métrica:
    - **Structure Similarity (SS)**

TPC-H

Resultados Execution Accuracy:

Prompt Engineering 

Modelo	Execution Accuracy (%)
DeepSeek: R1 Distill Llama 70B	4%
Google: Gemini 2.0 Flash Experimental	0%
Meta: Llama 3.3 70B	0%

Resultados TPC-H

Resultados Structure Similarity :

Modelo	Average Similarity (%)
DeepSeek: R1 Distill Llama 70B	67%
Google: Gemini 2.0 Flash Experimental	65%
Meta: Llama 3.3 70B	61%

Prompt Engineering

Iterative Refinement

Modelo	Average Similarity (%)
DeepSeek	65%
ChatGPT	81%
Gemini	82%

TPC-C

A descrição de cada transação, disponibilizada pelo documento do TPC-C, foi fornecida a modelos LLM, nomeadamente o ChatGPT, o DeepSeek e o Gemini.

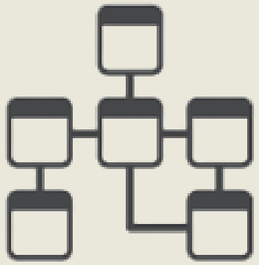
A transacção gerada foi depois avaliada através de structure similarity, uma vez que se espera que a query final não seja executável, dado que o modelo não possui conhecimento prévio do esquema da base de dados.

Foram utilizados dois métodos de avaliação: **single prompt e iterative refinement.**

Resultados

Modelo	Stock Level Transaction - SP	Stock Level Transaction - IR	Order Status Transaction - SP	Order Status Transaction - IR
GPT-5	86.36%	86.36%	37.12%	37.12%
Gemini 2.5	84.78%	84.78%	45.67%	45.67%
Deepseek V3.2	20.34%	54.24%	25.83%	37.30%

Context-Aware SQL Generation

Q. What is the total amount spent on orders? +  orders(id, customer_id, total_amount, date) → **A.**

```
SELECT
SUM(total_amount)
AS total_spent
FROM orders;
```

Prompt Engineering

Fornecer instruções detalhadas para guiar a LLM a produzir queries mais consistentes.

- **Zero-shot**

Schema: orders(id, customer_id, total_amount, date)

Question: What is the total amount spent on orders?

- **Few-shot**

Example:
Q: List product names with price above 100.
A: SELECT name FROM products WHERE price > 100;

Now:
Schema: orders(id, customer_id, total_amount, date)
Q: What is the total amount spent on orders?

Reasoning

Explorar as capacidades de reasoning das LLMs utilizando o chain-of-thought para gerar passos intermédios antes da resposta final.

- **Zero-shot**

Schema: orders(id, customer_id, total_amount, date)

Q: What is the total amount spent on orders?

A: Let’s think step by step.

- **Few-shot**

Example:
Q: List product names with price above 100.
A: SELECT name FROM products WHERE price > 100;
CoT: Let's think step by step.
According to "product names", columns [products.name] may be used.
...
So the final answer is:
SELECT name FROM products WHERE price > 100;

...

Escolha dos exemplos few-shot

Maximal Marginal Relevance (MMR):

$$\text{MMR\_score} = \lambda \times \text{relevance} - (1 - \lambda) \times \text{redundancy}$$

- Relevance: Similaridade entre exemplo candidato e query atual
- Redundancy: Similaridade do candidato com exemplos já selecionados
- λ (Lambda): Peso que equilibra similaridade vs. diversidade
  - Mais próximo de 1 → prioriza similaridade
  - Mais próximo de 0 → prioriza diversidade

- Seleciona iterativamente o exemplo com maior MMR_score
- Evita selecionar múltiplos exemplos semelhantes, garantindo que os exemplos escolhidos são relevantes mas não redundantes

Abordagem para o TPC-H

1. Executar queries originais TPC-H e guardar outputs de referência e tempos de execução
2. Executar queries geradas pelas LLMs e guardar os respetivos outputs e tempos de execução
3. Comparar resultados usando duas métricas:
    - **Execution Accuracy (EX):** avalia a correção funcional das queries geradas, verificando se produzem os mesmos resultados que as originais
    - **Valid Efficiency Score (VES):** avalia simultaneamente a correção e eficiência das queries, considerando o tempo de execução

TPC-H: Resultados

Zero-shot:

Modelo	Execution Accuracy (%)	Valid Efficiency Score (%)
Gemini	95,24	77,86
GPT	71,43	53,12
DeepSeek	52,38	40,57
Llama	52,38	35,06

Zero-shot + Reasoning:

Modelo	Execution Accuracy (%)	Valid Efficiency Score (%)
Gemini	95,24	79,66
GPT	61,90	50,72
DeepSeek	52,38	42,56
Llama	42,86	38,54

TPC-H: Resultados

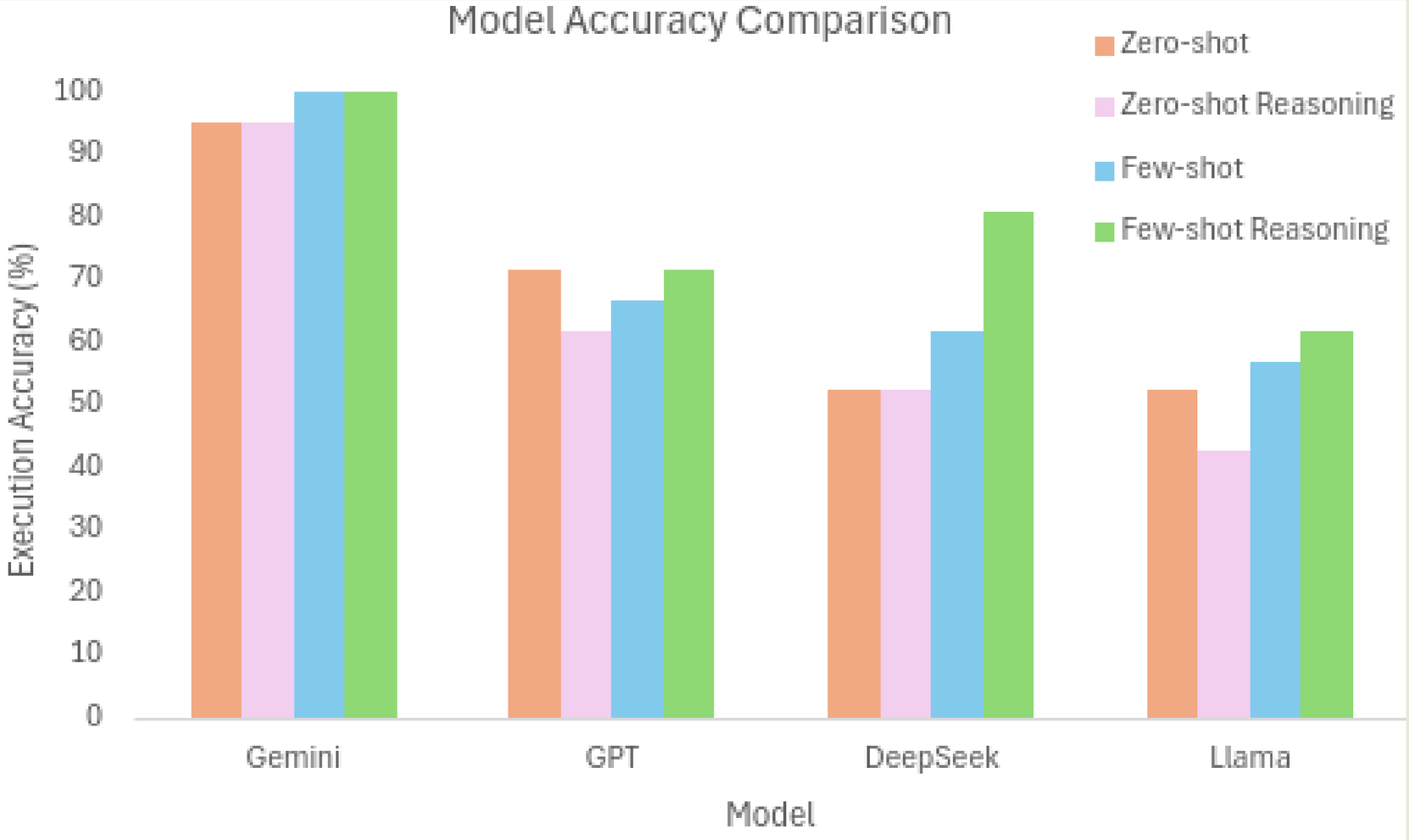
Few-shot:

Modelo	2Ex, λ=0.4		2Ex, λ=0.6		5Ex, λ=0.4		5Ex, λ=0.6	
	EX (%)	VES (%)	EX (%)	VES (%)	EX (%)	VES (%)	EX (%)	VES (%)
Gemini	95,24	84,76	100,00	79,00	90,48	73,14	85,71	67,49
GPT	57,14	49,09	61,90	51,15	66,67	48,44	66,67	53,18
DeepSeek	47,62	41,60	52,38	38,21	61,90	51,35	42,86	35,03
Llama	52,38	51,53	52,38	47,31	57,14	49,44	47,62	39,20

Few-shot + Reasoning:

Modelo	2Ex, λ=0.4		2Ex, λ=0.6		5Ex, λ=0.4		5Ex, λ=0.6	
	EX (%)	VES (%)	EX (%)	VES (%)	EX (%)	VES (%)	EX (%)	VES (%)
Gemini	100,00	82,40	90,48	95,29	100,00	80,56	95,24	76,06
GPT	57,14	46,79	66,67	55,52	61,90	42,75	71,43	54,99
DeepSeek	80,95	65,60	76,19	59,76	52,38	49,47	76,19	60,97
Llama	57,14	51,62	61,90	67,78	57,14	43,04	57,14	45,88

Análise dos Resultados TPC-H



Abordagem para o TPC-C

1ª Fase - Ambiente Controlado (8 execuções por transação):

- Capturar estado inicial com snapshots
- Executar transações originais e geradas
- Comparar outputs e estado final da BD

2ª Fase - Benchmark Completo:

- Executar benchmark TPC-C completo para avaliar desempenho em ambiente concorrente e com maior variedade de inputs
- Registar para cada modelo e método de prompting:
  - Latência média
  - Taxa de erro (%)
- Referência: Latência original = 0.005s

TPC-C: Resultados

Zero-shot:

Modelo	Execution Accuracy (%)	TPC-C Benchmark	
		Avg Latency (s)	Error Rate (%)
Gemini	60,0	0,0042	49,6
GPT	90,0	0,0052	2,1
DeepSeek	40,0	0,0048	73,5
Llama	20,0	0,0058	72,0

Zero-shot + Reasoning:

Modelo	Execution Accuracy (%)	TPC-C Benchmark	
		Avg Latency (s)	Error Rate (%)
Gemini	77,5	0,0054	66,5
GPT	92,5	0,0048	15,0
DeepSeek	60,0	0,0066	51,0
Llama	20,0	0,0046	74,0

TPC-C: Resultados

Few-shot:

Modelo	Execution Accuracy (%)	TPC-C Benchmark	
		Avg Latency (s)	Error Rate (%)
Gemini	95,0	0,0050	41,5
GPT	92,5	0,0056	17,7
DeepSeek	75,0	0,0046	60,1
Llama	30,0	0,0036	80,0

Few-shot + Reasoning:

Modelo	Execution Accuracy (%)	TPC-C Benchmark	
		Avg Latency (s)	Error Rate (%)
Gemini	97,5	0,0062	20,4
GPT	85,0	0,0082	16,3
DeepSeek	97,5	0,0060	41,7
Llama	40,0	0,0066	60,0

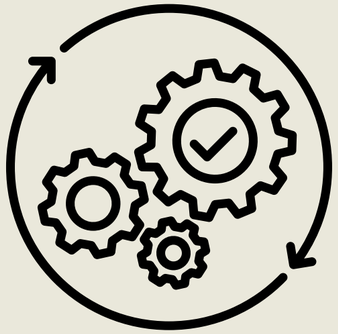
Análise dos Resultados TPC-C

- Os melhores resultados foram obtidos com os métodos few-shot
- Existe uma discrepância entre correção funcional (Fase 1) e robustez concorrente (Fase 2). Transações corretas em ambiente controlado falharam sob concorrência
- Nenhuma transação gerada igualou a performance das originais

Direct Data Access

Table Question Answering (Table QA) é uma tarefa de NLP que se concentra em fornecer respostas às consultas de utilizadores via compreensão e raciocínio sobre dados tabulares.

Q. What is the Total Amount Spent on orders ?



A. 500$

Direct Data Access

Atualmente, existem duas abordagens principais utilizadas para melhorar o desempenho de LLMs em Table QA:

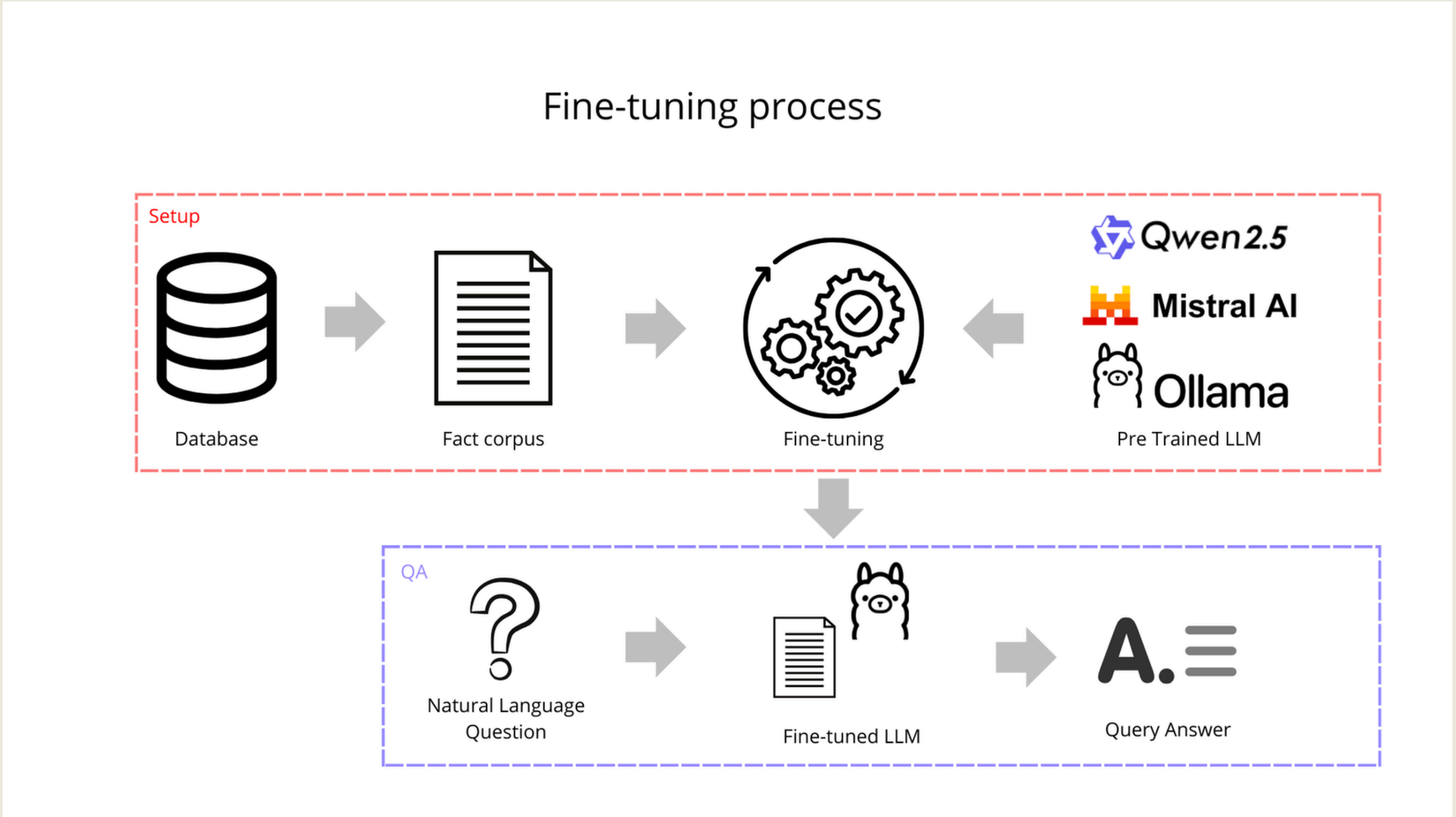
Domain-Specific Fine-Tuning (DSFT)

Melhora o desempenho das LLMs em QA, treinando diretamente os modelos no corpus específico da base de dados

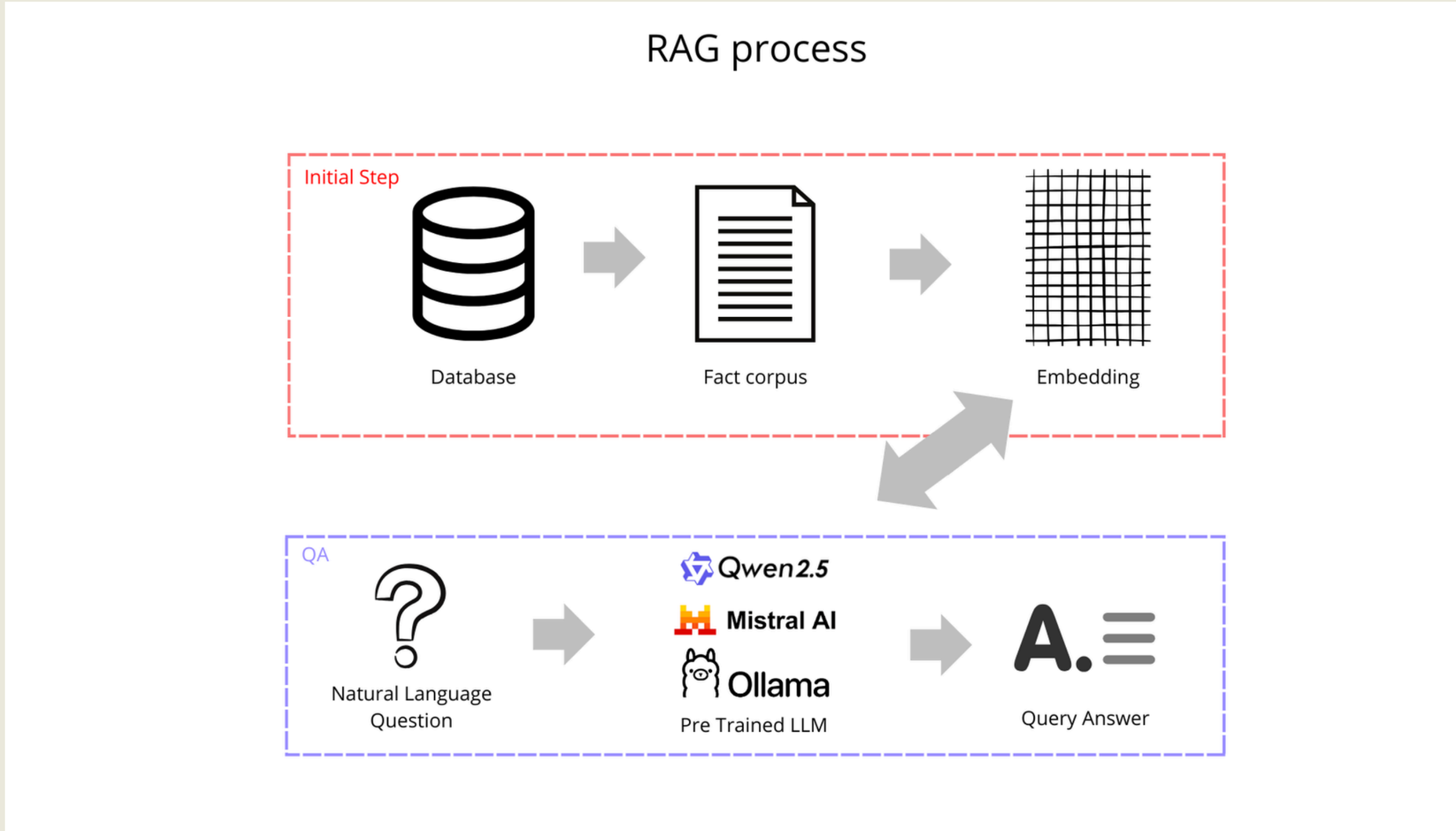
Retrieval-Augmented Generation (RAG)

Melhora o desempenho das LLMs em QA, usando o corpus específico de uma base de dados como fonte de conhecimento externa para dar suporte às respostas do modelo

Direct Data Access



Direct Data Access



Direct Data Access

Paradigma	Pros	Contras
DSFT	Incorpora conhecimento específico de uma base de dados diretamente nos parâmetros do modelo.	Requer muito poder de computacional e incorre em altos custos devido ao processo de treino.
RAG	Método de menor custo, pois não requer atualização dos parâmetros do modelo. Utiliza o corpus extraído como base de conhecimento externa, facilitando potencialmente a atualização do conhecimento.	A construção de uma base de dados vetorial exige recursos de memória substanciais. O desempenho depende da precisão da recuperação e da qualidade das representações semânticas nos blocos de texto.

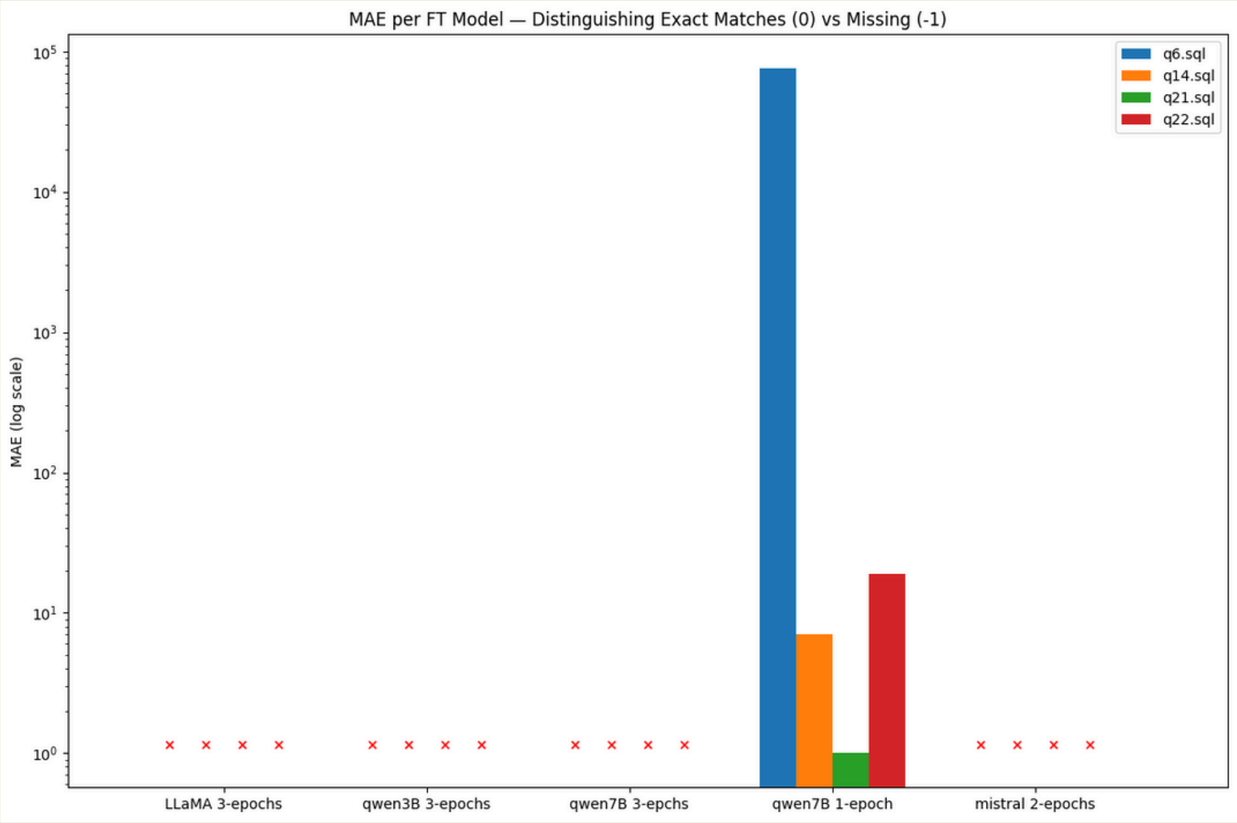
TPC-H

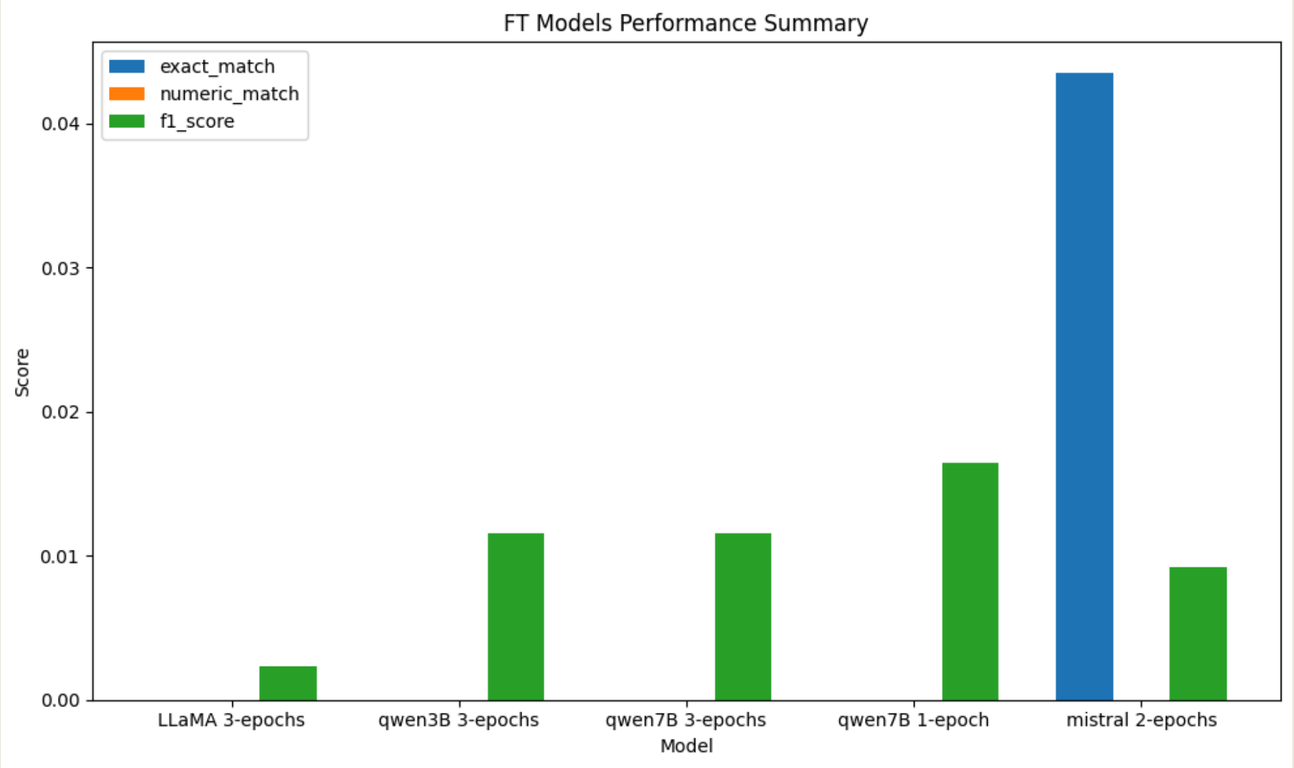
Setup

- Subset do TPC-H (≤10k linhas por tabela)
- Ground truth retirado após executar as queries sobre o subset
- Strict prompting: formatação de output entre outras regras para evitar alucinações

TPC-H

Fine-tuning





TPC-H

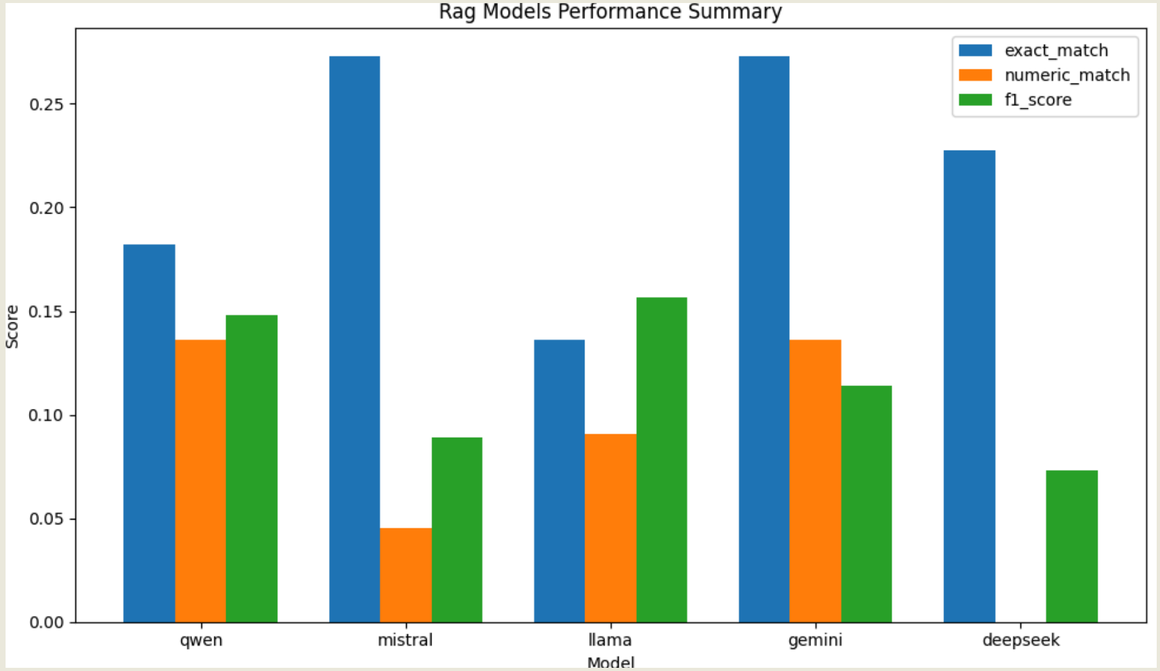
Fine-tuning

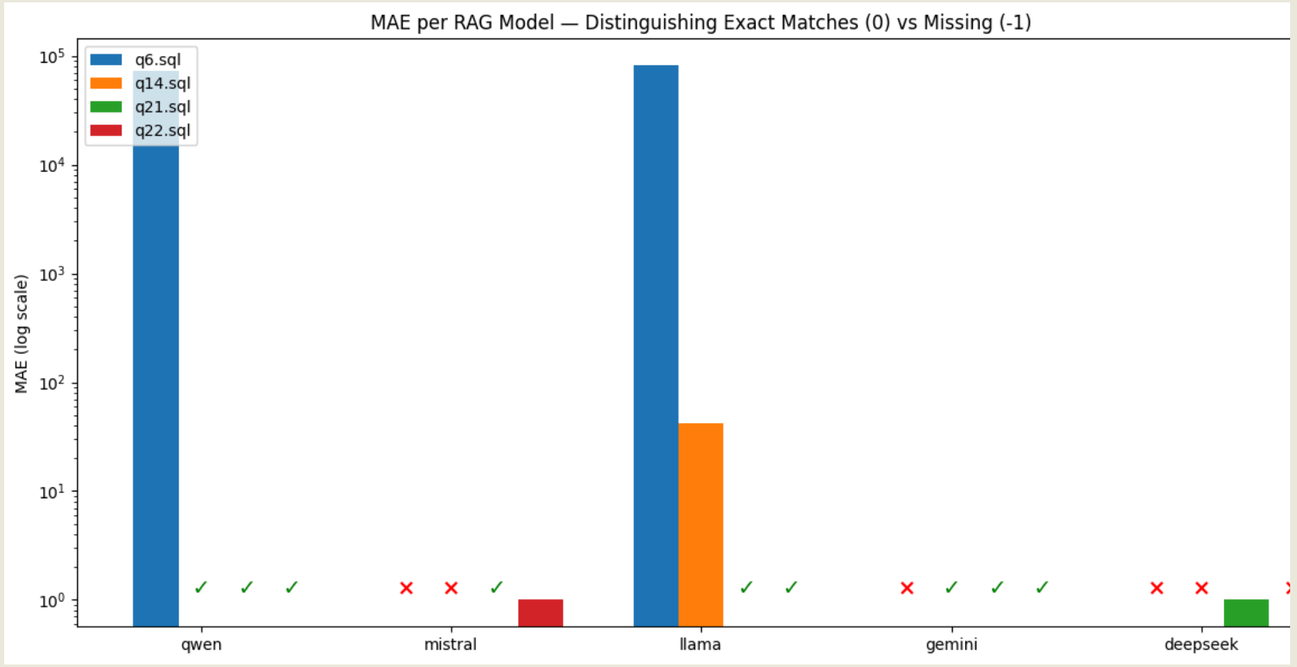
Por que é que fine-tuning é inadequado?

- Não tem capacidade para armazenar dados estruturados
- Incapaz de juntar e agregar dados
- Não permite mudanças sobre o corpo de factos
- Não é determinista nem verificável

TPC-H

RAG


Rag Models Performance Summary
exact_match
numeric_match
f1_score
Score
0.00
0.05
0.10
0.15
0.20
0.25
qwen
mistral
llama
gemini
deepseek
Model


MAE per RAG Model — Distinguishing Exact Matches (0) vs Missing (-1)
q6.sql
q14.sql
q21.sql
q22.sql
MAE (log scale)
10^0
10^1
10^2
10^3
10^4
10^5
qwen
mistral
llama
gemini
deepseek

TPC-H

RAG

Desvantagens

- Janela de contexto é limitadora, especialmente para benchmarks como o TPC-H
- Os modelos continuam a não conseguir fazer agregações.
- O processo de “Retrieve” não dá garantias sobre o raciocínio dos modelos.

Vantagens

- Bom com questões simples de consulta/”look up”.
- Consegue detetar quando os factos não são suficientes para responder a uma questão.

TPC-C

Características do TPC-C

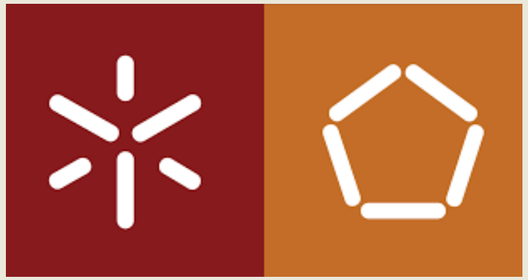
- O Benchmark TPC-C serve-se de processos transacionais, incluindo operações do tipo DML, que não são compativeis com LLMs.
- As Métricas principais do TPC-C, como o throughput (tpmC) ou a latência de execução.

Conclusão

Concluímos que os modelos dependem fortemente do conhecimento do esquema para gerar queries executáveis, embora consigam produzir resultados relativamente próximos do esperado. Observámos também limitações claras quando se tenta o acesso direto aos dados, especialmente no nível 3.

- Pontos fortes dos LLMs: interpretação de pedidos e geração de SQL.
- Limitações: não substituem motores de bases de dados para computação ou agregação.
- Futuro: arquiteturas híbridas, com LLMs como planeadores/tradutores e SGBD garantindo execução fiável, escalabilidade e precisão.
- Metodologia: mitigou variabilidade de respostas e limitações de benchmarks em configurações reduzidas.

Em suma, os tópicos propostos foram abordados e os resultados reforçam o papel complementar dos LLMs no contexto de bases de dados.



LLM-based Querying vs SQL

Eduardo Cunha PG55939
Jorge Rodrigues PG55966
Tiago Rodrigues PG56013
Vasco Faria PG57905