# Trabalho 4: Análise Quantitativa do Trade-off entre Especialização e Generalização em LLMs via Fine-Tuning

Acauan C. Ribeiro
Universidade Federal do Amazonas (UFAM), Manaus, Brasil
acauan.ribeiro@ufam.edu.br

***Resumo*—Este trabalho investiga empiricamente o impacto do fine–tuning com *Parameter–Efficient Fine–Tuning* (PEFT) LoRA sobre o modelo `Meta-Llama-3-8B-Instruct` na tarefa *Text–to–SQL*. Avaliamos o ganho de desempenho na tarefa-alvo (dataset Spider) e a variação de capacidade em tarefas de conhecimento geral (subset MMLU), quantificando o trade–off entre especialização e generalização. Resultados mostram um salto de +563% na *Execution Accuracy* para Text–to–SQL, sem evidências de *esquecimento catastrófico*: o modelo fine–tuned apresentou melhora de até +261% em MMLU. Discutimos implicações práticas, limitações e caminhos futuros.**

***Palavras-chave*—Large Language Models, Fine–Tuning, LoRA, Text–to–SQL, MMLU, Transfer Learning**

## I. Introdução

Os Modelos de Linguagem de Grande Porte (LLMs) têm demonstrado desempenho notável em tarefas diversas. Um desafio recorrente é equilibrar a *especialização* em uma tarefa específica com a *generalização* para domínios amplos. Este estudo analisa esse trade–off ao aplicar fine–tuning com LoRA [1] no `Meta-Llama-3-8B-Instruct` [2] para gerar SQL a partir de linguagem natural, uma tarefa avaliada pelo benchmark Spider [3].

## II. Metodologia

O pipeline experimental foi projetado para ser modular e reprodutível, consistindo em pré-processamento de dados, fine-tuning e avaliação em dois benchmarks distintos.

### A. Pipeline de Dados

- **Spider** (Text–to–SQL): Utilizamos 7.000 instâncias do *training split* para o fine–tuning e 1.034 instâncias do *development split* para avaliação da tarefa–alvo.
- **MMLU Subset**: Para avaliar a generalização, criamos uma suíte de 150 questões do MMLU [4], balanceadas igualmente em **três** macroáreas (STEM, Humanidades e Ciências Sociais), conforme especificado.

**Recursos Disponíveis:** O código-fonte, os scripts e os resultados deste projeto estão disponíveis no GitHub: https://github.com/acauanrr/nlp_trab4_tradeoff.git. O fluxo de trabalho completo pode ser executado através do notebook Google Colab: https://colab.research.google.com/drive/1EOtC8O84xTV0062KKlmNGfNHs5b_WbFV.

### B. Configuração do Fine–Tuning

O fine–tuning foi realizado com o `SFTTrainer` da biblioteca TRL da Hugging Face, empregando duas configurações LoRA (Tabela I). Apenas $0.52\,\%$ e $1.03\,\%$ dos parâmetros totais do modelo foram treinados, respectivamente, demonstrando a eficiência do método.

```
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16,
)
model = AutoModelForCausalLM.from_pretrained(
    model_id,
    quantization_config=bnb_config,
    device_map="auto",
)
```

Listagem 1. Configuração QLoRA (4-bit) em `train_lora.py`

TABELA I
Hiperparâmetros das Configurações LoRA

| **Config.** | $r$ | $\alpha$ | Dropout | LR | Steps |
|---|---|---|---|---|---|
| LoRA–1 | 16 | 32 | 0.05 | $2 \times 10^{-5}$ | 2048 |
| LoRA–2 | 32 | 64 | 0.1 | $1 \times 10^{-5}$ | 2048 |

### C. *Métrica* Execution Accuracy

Implementamos a métrica *Execution Accuracy* no DeepEval (`custom_metrics/execution_accuracy.py`), que executa a query gerada sobre a base de dados e compara o resultado com a referência. A Listagem 2 detalha a lógica.

### D. Disponibilidade e Reprodutibilidade

Para garantir a total reprodutibilidade deste estudo, todo o código-fonte, scripts, notebooks de execução e resultados brutos foram disponibilizados publicamente no repositório do projeto.

## III. Resultados

### A. Desempenho na Tarefa Text–to–SQL

A especialização via LoRA gerou um ganho de performance expressivo, como detalhado na Tabela II.

```python
def measure(self, test_case: LLMTestCase) -> float:
    db_path = os.path.join(self.db_root_path,
                           test_case.context[0],
                           f"{test_case.context[0]}.
    sqlite")

    pred_res, pred_err = self._execute_sql(
        test_case.actual_output, db_path)
    if pred_err:
        self.reason = f"Erro na query gerada: {
    pred_err}"
        return 0.0

    exp_res, gt_err = self._execute_sql(
        test_case.expected_output, db_path)

    if pred_res == exp_res:
        self.score = 1.0
    else:
        self.score = 0.0

    return self.score
```

Listagem 2. Lógica principal da métrica Execution Accuracy

TABELA II
RESULTADOS NO SPIDER DEV SPLIT

| **Modelo** | **Execution Accuracy (%)** |
|---|---|
| Baseline (Few-shot) | 9.17 |
| Fine-tuned (LoRA–1) | 60.83 |
| Fine-tuned (LoRA–2) | 60.83 |

### B. Desempenho em Conhecimento Geral (MMLU)

A Tabela III apresenta a acurácia no MMLU. Contrariando a hipótese de esquecimento, ambos os modelos fine-tuned superaram o baseline.

TABELA III
RESULTADOS NO MMLU SUBSET (ACURÁCIA)

| **Modelo** | Overall | STEM | Humanidades | Sociais |
|---|---|---|---|---|
| Baseline | 8.67% | 0.00% | 26.00% | 0.00% |
| LoRA–1 | **31.33%** | **34.00%** | 58.00% | 2.00% |
| LoRA–2 | 24.67% | 8.00% | **64.00%** | 2.00% |

### C. Análise do Trade–off

A Tabela IV quantifica a variação percentual de desempenho, evidenciando um ganho simultâneo em ambas as frentes.

TABELA IV
VARIAÇÃO DE PERFORMANCE VS. BASELINE

| **Modelo** | Δ Spider | Δ MMLU (Overall) | Δ MMLU (Human.) |
|---|---|---|---|
| LoRA–1 | +563% | **+261%** | +123% |
| LoRA–2 | +563% | +185% | +146% |

## IV. ANÁLISE DE ERROS

A Tabela V ilustra três categorias de falhas observadas no modelo Fine-tuned 1 no dataset Spider, o que ajuda a identificar suas limitações residuais.

## V. DISCUSSÃO

### A. Ganho de Especialização sem Perda de Generalização

Os resultados indicam que o fine–tuning com LoRA proporcionou um ganho substancial na tarefa específica (+563% de *Execution Accuracy*) enquanto também elevou o desempenho geral em MMLU, contradizendo a expectativa de um trade–off negativo.

**Hipótese 1 – Robustez do Llama 3:** A forte capacidade de raciocínio do modelo base pode ter sido reforçada pelo treinamento em SQL, melhorando a inferência lógica geral, que é testada em MMLU.

**Hipótese 2 – Natureza do LoRA:** Ao atualizar uma fração mínima dos parâmetros (menos de 1.1% neste estudo), o método PEFT preserva a vasta maioria do conhecimento pré-treinado, mitigando o risco de *esquecimento catastrófico*.

**Hipótese 3 – "Contaminação" Positiva do Dataset:** O corpus Spider, embora focado em SQL, contém linguagem natural diversificada e complexa nas perguntas, o que pode ter servido como sinais de treinamento auxiliares para o conhecimento geral do modelo.

### B. Comparação das Configurações LoRA

Ambas as configurações atingiram a mesma performance máxima no Spider (60.83%), sugerindo que $r$=16 já é suficiente para capturar a complexidade da tarefa Text-to-SQL. Curiosamente, a configuração LoRA–1 ($r$=16) superou a LoRA–2 ($r$=32) no MMLU (31.33% vs. 24.67%), especialmente na categoria STEM. Isso pode indicar que uma adaptação menor e mais focada (menor ‘r‘) é mais eficaz em melhorar o raciocínio lógico sem "perturbar" as outras áreas de conhecimento do modelo.

### C. Implicações Práticas

Nossos achados sugerem que, para LLMs modernos e robustos como o Llama 3, a especialização em tarefas de raciocínio estruturado pode ser segura, ou até benéfica, no que tange à manutenção da capacidade de generalização, especialmente quando se utilizam técnicas PEFT. Isso encoraja a criação de agentes comerciais especializados sem o receio de sacrificar suas habilidades amplas.

## VI. CONCLUSÃO

Demonstramos empiricamente que o fine–tuning com LoRA no modelo `Meta-Llama-3-8B-Instruct` para a tarefa Text–to–SQL não apenas aumenta drasticamente a performance na tarefa específica, mas, de forma inesperada e positiva, também melhora a sua capacidade de resposta em tarefas de conhecimento geral. Futuras investigações poderiam explorar a escalabilidade deste fenômeno para outros domínios de tarefas e analisar mais a fundo o impacto do rank e de outros hiperparâmetros de LoRA neste trade-off positivo.

TABELA V
EXEMPLOS DE FALHAS DO MODELO FINE-TUNED 1 NO DATASET SPIDER

| **Pergunta** | **SQL Predito** | **SQL Esperado** | **Motivo da Falha** |
|---|---|---|---|
| How much does the youngest dog weigh? | `SELECT max(weight) FROM Pets WHERE PetType = 'dog'` | `SELECT weight FROM pets ORDER BY pet_age LIMIT 1` | **Erro Semântico:** O modelo confundiu "mais jovem" (`ORDER BY`) com "mais pesado" (`MAX()`). |
| Find the major and age of students who do not have a cat pet. | `... EXCEPT SELECT ...` | `... WHERE stuid NOT IN (SELECT ...)` | **Erro Lógico:** A query com `EXCEPT` falha, pois remove qualquer estudante que tenha um pet (não apenas gatos) de outra matéria, retornando um resultado incorreto. A subquery com `NOT IN` é a abordagem correta. |
| Find the average and maximum age for each type of pet. | `SELECT pettype, avg(pet_age), max(pet_age) ...` | `SELECT avg(pet_age), max(pet_age), pettype ...` | **Erro de Ordem:** A query é semanticamente correta, mas a ordem das colunas no `SELECT` difere da referência, fazendo com que a comparação de tuplas falhe na métrica de acurácia. |


## AGRADECIMENTOS

Agradeço aos docentes das disciplinas ICC220 e PPGINF528, Prof. Dr. André Carvalho e Prof. Dr. Altigran da Silva, pela orientação e pelo design desafiador do projeto.


## REFERÊNCIAS


[1] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021.

[2] AI at Meta, "The llama 3 herd of models," https://ai.meta.com/blog/meta-llama-3/, 2024.

[3] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman *et al.*, "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*, 2018, pp. 3911–3921.

[4] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2021.